

Abstract

A new word processing program, Rewrite, universally compatible with all Atari 8-bit computer systems, has been designed especially for teachers. During a two-year period, twenty-five high school teachers were consulted to determine the difficulties encountered in educational word processing. Nine popular word processors were chosen for intensive testing and evaluation to aid in creating one more suitable to the needs of teachers than those currently available. Of those evaluated, Rewrite has twice as many features desired by teachers and twenty percent more capabilities overall.

Narrowing the Gap Between
the Word Processing Needs of Teachers
and
the Capabilities of Word Processors for
Atari 8-bit Computers

by
Robert Ertl

Submitted in partial fulfillment of the requirements
for the degree of Masters in Education
at Carthage College

Kenosha, Wisconsin

July 1987

Acknowledgments

Trade Marks

400, 800, 600XL, 800XL, 1200XL, 65XE, and 130XE are registered trade marks of Atari Corporation, Sunnyvale, CA

AXLON and Ramdisk are registered trademarks of Axlon Inc., Sunnyvale, CA

Table of Contents

Abstract.....	i
Title Page.....	ii
Acknowledgments.....	iii
Table of Contents.....	iv
List of Tables.....	vii
List of Figures.....	viii

Chapter One

Introduction

Significance of the Study.....	1
Statement of the Problem.....	2
Statement of Hypotheses.....	3
Hypothesis one.....	3
Hypothesis two.....	3
Hypothesis three.....	4
Rational for the Choice of Hypotheses.....	4
Computer Related Terms.....	4

Chapter Two

Review of the Literature

Word Processing Needs and Capabilities.....	6
Development of a Word Processor.....	8

Chapter Three

Method

Overview.....	9
The Setting.....	9
Assessment of the Word Processing Needs.....	10
The sample.....	10
The means of data collection.....	10
Assessment of the Capabilities of Word Processors...	11
The sample.....	11
The new word processor.....	11
The instrument.....	12
The method of analysis.....	13

Chapter Four

Review of the Unquantified Data

Difficulties With Text Documents.....	14
Indenting.....	16
Outdenting.....	16
Sequentially numbered items.....	17
Alignment of items on different lines.....	17
Special symbols and diagrams.....	18
Duplicating masters.....	19
Difficulties With Data Files.....	19
Difficulties With Time Requirements.....	20
Difficulties Caused by Lacking Information.....	21
Difficulties Caused by Incompatibility.....	21

Chapter Five	Analysis and Interpretation of the Data	
Hypothesis One.....		24
Hypothesis Two.....		24
Hypothesis Three.....		28
The first requirement for acceptance.....		28
The second requirement for acceptance.....		33
 Chapter Six	 Conclusions and Applications	
Features Not Implemented in <u>Rewrite</u>		35
Unnecessary duplication avoided.....		35
Incompatible features.....		37
Complex features.....		38
Applications.....		39
 References.....		40
 Appendix A		
Glossary of Computer Terms.....		43
 Appendix B	 Word Processor Evaluation	
Results.....		47
Descriptions of the Criteria Used.....		52
 Appendix C		
<u>Rewrite</u> Reference Manual.....		65

List of Tables

Table 1

Word Processing Capabilities Which Can Help Satisfy
The Currently Unsatisfied Needs of Teachers.....25

Table 2

Assessment of Ability to Satisfy the Currently
Unsatisfied Word Processing Needs of Teachers.....27

Table 3

Overall Assessment of Word Processing Capabilities..34

List of Figures

Figures 1

Examples of text document components.....15

Figure 2

Printer font of mathematics symbols.....32

Chapter One

Introduction

Significance of the Study

In 1979 Atari released its original 400 and 800 model computers. The first word processors for these computers became available in 1981 and were received as marvelous innovations. Many of them, however, had a limited number of word processing capabilities and some could only be used with specific printers which are no longer available.

Since 1983 the 600XL, 800XL, 1200XL, 65XE, and 130XE models have been released with enhanced hardware capabilities and restructured operating systems. Many of the early word processors are not compatible with the new computer models due to the way in which they were interfaced with the original operating system.

It was not until 1985 that word processors became available which were compatible with all of the Atari computers and which were easily adaptable to a variety of printers. While these word processors were designed to have greater potentialities than their predecessors, some of the new features could not be utilized fully due to programming errors. While these problems have nearly all

been corrected in current revisions there is still a disparity between the word processing capabilities that are available and those that are needed. Specifically, many word processing needs of teachers remain unsatisfied.

Statement of the Problem

This study focused on the word processing needs of teachers at J. I. Case High School in Racine, Wisconsin. The scope of this study was limited to word processors for the Atari 8-bit computer, since it was the most readily available computer system at J. I. Case when this study began.

The need for this study first became evident in the fall of 1983 when many of the teachers who were using the Atari computers for word processing began to seek assistance from this researcher for the difficulties they were having. Occasionally, the problems which had arisen were due to the teacher's lack of understanding regarding the operations of the word processor being used. More frequently, however, the word processor in question was not capable of producing the desired results, or the desired results could be obtained only through some time-consuming and/or difficult process. When the word processor's inadequacy was the source of a teacher's difficulty, the teacher either manually augmented the word processor's output, settled for results that were of lesser quality than desired, completed the task by some other means, or

decided not to complete the task.

Quite often, the specific difficulty encountered by a teacher, could have been avoided through the use of a different word processor. Some of the teachers, however, were unwilling to use more than one word processor for their work. These teachers insisted that if they were going to invest the time necessary to learn how to use a new word processor, it would have to be capable of being used for all of their current word processing tasks.

When this study began, no single word processor was known to this researcher that was capable of satisfying the needs that were demonstrated by the difficulties the teachers were having. Also, a few of the teachers' needs could not be satisfied adequately by any of the word processors known to this researcher. Yet, it was felt that it should be possible to implement a word processor on the Atari computer that would alleviate the majority of the problems for which the teachers were seeking assistance.

Statement of Hypotheses

Hypothesis one.

Teachers have word processing needs that are not currently being satisfied by the word processing programs they are using.

Hypothesis two.

No word processor has yet been implemented on the Atari computer which has more than three-fourths of the

capabilities necessary to meet the currently unsatisfied word processing needs of teachers.

Hypothesis three.

It is possible to design and implement a word processing program for the Atari computer which has more of the capabilities necessary to meet the currently unsatisfied word processing needs of teachers, and more capabilities overall, than any of the popular word processors available for this computer.

Rational for the Choice of Hypotheses

These hypotheses were chosen to guide the course of this study. If no evidence is found which favors the acceptance of hypothesis one, this study should be terminated. The acceptance of hypothesis one, however, will firmly establish the need for this study. If evidence is found which supports the rejection of hypothesis two, this study should be concluded with recommendations to teachers regarding word processors for the Atari computer. If hypothesis two is accepted, this study should be continued in an attempt to satisfy hypothesis three.

Computer Related Terms

It is difficult to accurately describe the specific details of the subject of word processing without using many of the computer related terms which have only recently been given their present-day meaning. The language of

computer science has been growing so rapidly that even the most recently published dictionaries of computer terms are often inadequate. For this reason a glossary of all such terms used in this paper has been included in Appendix A.

Chapter Two

Review of the Literature

In order to assess how well the word processing needs of teachers were being met by the word processing programs that were available for the Atari computer, two types of information had to be obtained. First, the currently unsatisfied word processing needs of teachers had to be determined. Second, the popular word processing programs for the Atari computer had to be identified, and the exact capabilities of these programs had to be ascertained.

To actually design and implement a word processing program, two other types of information had to be obtained. First, it was necessary to have details about the hardware design, the various general operating systems, and the disk operating systems of the Atari computer. Second, specific techniques for writing a word processor in assembly language were needed.

Word Processing Needs and Capabilities

There is virtually no literature available that specifically addresses the unique word processing needs of teachers. The word processing literature nearly always

expounds on what is available and possible, rather than on what is needed.

There were two types of published literature that were helpful in screening the word processing programs currently available: software reviews and software comparisons. Articles by Basovski (1986), Roquemoire (1986), Ciraol & Friedland (1985), Brannon (1985b), Riggs (1985), Leyenberger (1985), Leyenberger (1984), Lyons (1984), and Kushner (1983) were reviewed and the following six word processors were chosen for further study: Letter Perfect (LJK Staff, 1981), Atari Writer (Atari Staff, 1982), The Writer's Tool (OSS Staff, 1984), Paperclip (Ahlstrom & Moore, 1985), Atari Writer Plus (Atari Staff, 1985), and Speed Script (Brannon, 1985).

The appropriate reference manuals were used to identify and examine the capabilities of each word processor studied. Some of the features described in a few of the reference manuals cannot actually be implemented by the word processor. For example, when the procedures for double column printing described in the Paper Clip reference manual (Harding, 1985, p. 82) are followed, everything appears to work properly except that one line of source text is not printed between the end of the first column and the beginning of the second column on every page but the first. Also, features were found in some word processors that were not documented anywhere in the appropriate reference manual. For example, Text Wizard has

a word counting capability that is not documented in its reference manual.

Development of a Word Processor

The hardware and operating system theory of operation manuals (Atari, 1980a, 1980b) provided the overview of the Atari computer that was needed. The source code listings for the operating system (Atari, 1981), the 2.0S disk operating system (Wilkinson, 1982), and the disk utilities package (Atari, 1981), provided most of the detailed information that was necessary. Mapping the Atari (Chadwick, 1983) proved to be a valuable resource for clearing up the seemingly endless mysteries that arose while using the more technical references just mentioned.

Atari BASIC (Wilkinson, 1983), 6502 Assembly Language Subroutines (Leventhal & Saville, 1982), and Speed Script Word Processor (Brannon, 1985a) provided the needed examples of programming techniques required for producing a word processor. Programming the 6502 (Zaks, 1980) provided the information necessary for assembly language programming.

Chapter Three

Method

Overview

The immediate goal of this study was to provide solutions for the problems the teachers were having when using the Atari computer systems for word processing. Three phases of original research were conducted in an effort to achieve this goal. First, information was gathered regarding the teachers' word processing difficulties. Second, the capabilities of nine popular word processing programs for the Atari computer were identified and tested. Third, the results of these two phases were utilized to design and implement a new word processing program with many of the capabilities of the word processors studied as well as additional capabilities which offered solutions to the problems the teachers were having.

The Setting

This study was conducted at J. I. Case High School in Racine, Wisconsin, beginning in the spring of 1985. The total number of Atari computers at the school varied

between 30 and 40 during the study. Slightly over half of the computers were the original 800 models. There were also two 600XL models and one 130XE model. The rest were 800XL models. Approximately one-third of all the computers made use of color monitors while the rest employed color television sets. There were 12 printers of various makes and models in use.

Assessment of the Word Processing Needs

The sample.

This study was inspired by the fact that twenty-five of the teachers at J. I. Case High school, who were using Atari computers for word processing, sought out this researcher for assistance when they encountered difficulties. These teachers all had an interest in the goal of this study and freely offered information regarding their word processing needs. Included were teachers of Business Education, Computer Science, English, Exceptional Education, Mathematics, and Social Studies.

The means of data collection.

When these teachers sought assistance, solutions were first explored within the limitations of the word processor being used. If appropriate, alternative solutions were considered which made use of different word processors. Finally, if the problem remained unresolved or if the best solution found was not easily carried out, attempts were made at conceiving new word processing features which could

satisfactorily solve the problem. No quantitative method of analysis was employed on this information. All word processing difficulties expressed were considered worthy of solution. A summary of the difficulties which the teachers encountered is given in the next chapter.

Assessment of the Capabilities of Word Processors

The sample.

The teachers were using the following word processors: Atari Word Processor (Atari Staff, 1981), Text Wizard (Robinson, 1981), Letter Perfect (LJK Staff, 1981), Atari Writer (Atari Staff, 1982), and Atari Writer Plus (Atari Staff, 1985). Also, some were using Bank Street Writer (Smith, Kusmiak & Olson, 1982) in their classrooms for instruction in word processing. These were all included in the sample of word processors that were to be evaluated. The additional word processors included in the sample, because of positive published reviews, were: The Writer's Tool (OSS Staff, 1984), Paperclip (Ahlstrom & Moore, 1985), and Speed Script (Brannon, 1985a).

The new word processor.

During this study a new word processing program entitled Rewrite (Ertl, 1987) was designed and implemented on the Atari computer by this researcher. It was written entirely in assembly language using the MAC/65 macro assembler (Lawrow, 1982), an Axlon 128K ramdisk, and an Atari 800 computer. Early working versions of Rewrite were

also used to edit the source code for this word processor in the latter stages of its development.

The design objectives were to provide the commonly available word processing features together with capabilities needed to meet the unsatisfied word processing needs of teachers in an easy to use and universally compatible package. During the actual programming of the word processor, efforts were continually made to produce efficient code both in terms of memory utilization and processing time. Subroutines were used extensively. The final version occupies 14K of RAM and was assembled from 152K of source code.

The instrument.

A list of word processor evaluation criteria (WPEC) was developed using four types of input. First, an initial list of specific word processing features and capabilities was compiled using the published reviews cited in the last chapter. Second, a preliminary examination of each of the word processors in the sample was conducted. As new word processing features and capabilities were found during these examinations, they were added to the list. Third, the word processing capabilities identified as possible solutions to the difficulties the teachers were having were included. Fourth, useful word processing capabilities which were conceived during the development of Rewrite were also included. The final version of the list of WPEC had 150 items which were organized into 19 categories. A

complete listing and a brief description of each criterion can be found in Appendix B.

The method of analysis.

Rewrite and the word processors in the sample were evaluated using the final version of the WPEC list and an Atari 800 computer system with 48K of RAM. An Atari 800 computer with 32K of RAM, an 800XL computer, and a 130XE computer were also used to make the necessary compatibility checks. The results were stored in a computer data file. The file was then analyzed with a computer program, designed and written by this researcher, to determine the following statistics for each word processor:

- * The number and percent of favorable evaluations received among the 30 items determined to be helpful in meeting the currently unsatisfied word processing needs of the teachers.
- * The number and percent of favorable evaluations among all 150 word processing evaluation criteria.

Chapter Four

Review of the Unquantified Data

This chapter contains a review of the word processing difficulties the teachers were having. As each is explained, the word processor evaluation criterion (WPEC) which is associated with a solution to the problem is referenced using the number it is identified with in Appendix B.

Difficulties With Text Documents

The basic components of the text documents, that the teachers were attempting to produce, were: left-justified blocks, indented blocks, outdented blocks, numbered blocks, and lines containing multiple items. An example of each of these components is given in Figure 1.

Many of the difficulties were linked to the fact that the screen editors of the word processors seldom displayed the text exactly as it would be printed. The discrepancy was frequently associated with inaccurate line parsing (WPEC #18). Screen editors with fixed line lengths were capable of showing proper line parsing only when the selected margins matched the fixed length of the editor

This is an example of a left-justified block of text. All of the lines in this block begin at the same horizontal print position.

This is an example of an indented block of text. The first line of this block begins five characters to the right of all subsequent lines in the block. This is also known as paragraphed text.

Outdented block: This is an example of an outdented block of text. The first line begins to the left of all subsequent lines in the text.

1. This is an example of a numbered block of text. It is actually a special form of an outdented block where the outdented part of the first line consists of a number followed by a period and two spaces.
2. This is an example of a numbered block of text that is followed by a line containing multiple items. This format is frequently used for multiple choice or multiple part exercises.
 - a. First choice b. Second choice c. Third choice

3a. First part

3b. Second part

Figure 1. Examples of text document components.

(WPEC #17). Most of the editors which could be adjusted to use different line lengths did not make use of the margin settings. This caused inappropriate line parsing to be displayed when the chosen line length was not consistent with the selected margins (WPEC #19). Other causes for inaccurate displays are described in many of the explanations of specific problems which follow.

Indenting.

Indented blocks of text were implemented by inserting spaces or by marking the first line of each paragraph with a special character and specifying the size of indent desired. When the marking method of indenting paragraphs was used, the actual indent was not displayed on the screen. As a result the lines were not always parsed as they would be when printed (WPEC #114).

Outdenting.

All of the teachers found outdented and numbered blocks of text to be either very tedious or difficult to produce. Some manually ended each line and began subsequent lines with the appropriate number of spaces. This method had the advantage of easy implementation as well as accurate screen displays. It did not, however, allow for easy subsequent insertion or deletion of text. It also resulted in very inefficient use of the text memory and disk space because of the many extra spaces that had to be used.

Others implemented outdented blocks by adjusting the

left margin setting before and after the first line of each block. While this method did not inhibit subsequent editing or require extra spaces, it was not as easy to implement as the first method and the outdented format was never accurately displayed by the screen editor. (WPEC #116).

Sequentially numbered items.

Different versions of exams and quizzes were sometimes needed. One method commonly used to produce a new form from an existing one was to rearrange the numbered items. These items then, had to be renumbered sequentially. This final tedious process could have been avoided if the word processor used was capable of numbering items automatically (WPEC #129).

Alignment of items on different lines.

The teachers often desired to arrange information in columns on the printed page. This was essential for the production of tables and charts. It was also desired for exercises with multiple choices or parts. Selectable tab stops (WPEC #49) and cursor column indicators (WPEC #14), when available, aided the initial positioning of text into columns but verifying this alignment visually was difficult for the following reasons:

1. Most of the teachers found it difficult to determine by visual means, whether or not one character on their screen was located directly beneath another character. A means of making such visual judgments quickly

and accurately was desired (WPEC #15).

2. The screen editors of all the word processors used by the teachers made use of word wrap. Word wrap is the process of parsing a body of text into lines without splitting words at the end of any line. Unfortunately, when the length of the printed line was greater than the maximum size the screen editor could display as a single line (WPEC #20), the word wrap feature distorted the relative positioning of the items which were "wrapped" onto subsequent lines (WPEC #39). Successful editing of existing text into columns under these circumstances required a meticulous counting of characters. This was complicated by the fact that none of the word processors in use distinguished between actual spaces and the false spaces produced by the word wrapping process (WPEC #16).

3. Imbedded codes usually had to be used to effect exponents, subscripts, special types of print, and/or underlining. Because these codes were displayed by the screen editors, the apparent horizontal position of the text which followed these codes was not accurate (WPEC #122 & #124).

Special symbols and diagrams.

The documents produced by some of the teachers (especially in the area of mathematics) were to contain special symbols and diagrams. These items had to be added manually to the printed page produced by the word processors. A means of producing these items with a word

processor was desired (WPEC #142).

Duplicating masters.

The teachers frequently wanted their text documents produced on some type of duplicating master. With multiple page documents it was necessary to insert a new master into the printer for each page of the document. This was most easily accomplished with some printers by turning them off, which also caused the loss of previously set initializations, such as commands for emphasized and/or double strike print, which were often necessary to produce good quality masters (WPEC #133).

Difficulties With Data Files

Data files were used to store: student records and scores for use with grading programs to determine grades and to print progress reports, responses to tests and questionnaires which were to be analyzed by statistical programs, source text for a specialized form printing program, and programming language source code which was to be assembled or compiled. All of the difficulties which accompanied attempts to align text items into columns, except having to deal with imbedded control codes, recurred in this setting. Difficulties also arose when the word processors used to generate or edit these files either did not permit all of the necessary characters to be entered (WPEC #31) or altered certain characters in the data when the files were saved (WPEC #90).

When records in the file were not supposed to be terminated with return characters (as was true with one of the grading programs) editing became very difficult. One method successfully used to work with such files was to terminate each field and record with a return character in a working copy of the file and then save both it and a version which had all of the return characters removed. Unfortunately, since the word processor being used did not permit a global removal of return characters (WPEC #66 & #72), this proved to be a tedious solution at best. These files would have been much easier to work with if the word wrap feature could have been disabled (WPEC #39) so that each record of the file could be displayed on a separate line (WPEC #20).

Difficulties With Time Requirements

A common complaint among all the teachers was that it usually took longer to produce the results they desired than they felt should be necessary. Two potential means of improving time efficiencies were identified.

It was observed that, when editing, the teachers usually spent more time moving the cursor than effecting changes. While some quick cursor movement commands were available on each of the word processors used, most of the teachers made regular use of only those commands which moved the cursor one character or line at a time. The automatic key repeat feature of the Atari computer (which

effects the repeat of any key 10 times each second after it is held down for more than one-half second), was usually taken advantage of to facilitate extended cursor movements. Improved editing efficiency could therefore be effected by increasing the key repeat rate (WPEC #33).

The teachers sometimes preferred not to have to horizontally scroll across the text of a document while proofreading. This required a lengthy reformatting process with one word processor. With another, the file had to be saved and then loaded again after the new line length was set. Similar operations had to be executed to restore the text to its original format. A quick method of entering and exiting a proofreading environment would make editing more time efficient (WPEC #21).

Difficulties Caused by Lacking Information

When the teachers sought assistance regarding the operations of a word processor, they rarely had the reference manual for the word processor they were using available at the school. The ability to view help files while editing a file (WPEC #97) would have been useful at these times. The availability of simple menus (WPEC #6) for carrying out basic operations would have also been helpful.

Difficulties Caused by Incompatibility

Two differences between the Atari 800 computer and the

newer XL and XE model computers caused some difficulties. First, the BASIC programming language is built-in on the XL and XE computer models while an external ROM cartridge is required for the 800 model. XL and XE BASIC can be disabled manually by holding the option key down when the power is turned on. When this was not done the available memory was reduced by 8K which led to: being unable to create or load large files, losing the ends of large files or having to reload the word processor. These problems could have been avoided if the word processor had disabled BASIC automatically (WPEC #3). Second, there were several improvements made in the XL and XE operating systems over the original. As a result, some of the word processors designed for the 800 model did not function properly on the XL and XE models. This restricted the accessibility to usable hardware for teachers who used these programs (WPEC #2). Another factor which affected hardware availability was that some of the 800 model computers had only 32K of RAM while some of the word processors required 48K (WPEC #5).

One-third of the televisions had overscan problems. This means that the extreme left and/or right columns of the standard text screen were not visible, making it very difficult to use those televisions with any word processor requiring a full screen. Word processors should be designed so that they can work without making use of these extreme text columns (WPEC #8). They should, however, also

be able to make full use of screens which do not have overscan problems (WPEC #9).

Some of the word processors were designed for exclusive use with printers which were no longer in use at the school and no means was available for configuring these programs for other printers (WPEC #1). As a result, features such as underlining, bold print, and exponents could not be used.

Chapter Five

Analysis and Interpretation of the Data

The observations made of the word processing difficulties the teachers were having, together with the evaluations of Rewrite and the nine word processors in the sample, were sufficient to test the hypotheses. Each of the hypotheses is restated below. An analysis of the relevant data follows each hypothesis.

Hypothesis One

Teachers have word processing needs that are not currently being satisfied by the word processing programs they are using.

Table 1 contains a list of the 30 word processing capabilities observed to be needed by teachers, but which were not supported by the word processors being used.

Hypothesis one is therefore accepted.

Hypothesis Two

No word processor has yet been implemented on the Atari computer which has more than three-fourths of the capabilities necessary to meet the currently unsatisfied word processing needs of teachers.

Table 2 shows the percentage of the items listed in

Table 1

Word Processing Capabilities Which Can Help Satisfy
The Currently Unsatisfied Needs of Teachers

No.	Word Processor Evaluation Criterion
1	Compatible with most printers
2	Compatible with 800, XL & XE models
3	Disables XL & XE BASIC
5	32K of RAM is sufficient
6	Menus with one or two levels
8	Screen width used is adjustable
9	Screen width can be 40 characters
14	Screen column indicator is available
15	Other visual alignment aid is provided
16	False spaces can be displayed
17	Scrolling screen editor is available
18	Editor displays accurate line content
19	Editor is adjusted with margin settings
20	Editor line length can be up to 200
21	Quick entry to and exit from proofreading

Table 1 continued

No.	Word Processor Evaluation Criterion
31	All ASCII characters can be entered
33	Fast automatic key repeat is available
39	Word wrap can be turned off
49	Cursor can be moved to selectable tab stops
66	Global search and replace is available
72	Search for return character is possible
90	No ASCII characters are altered
97	Help files can be viewed without losing text
114	Automatic indenting (displayed accurately)
116	Automatic outdenting (displayed accurately)
122	Underlining without delimiters
124	Superscripts & subscripts without delimiters
129	Automatic item numbering
133	Printer can be initialized with each page
142	Printer code files can be included

Note. A more detailed description of these word processing evaluation criterion (WPEC) can be found in Appendix B using the criterion numbers listed.

Table 2

Assessment of Ability to Satisfy the Currently
Unsatisfied Word Processing Needs of Teachers

<u>Name of the</u> <u>Word Processor</u>	<u>Number of</u> <u>WPEC Passed</u>	<u>Percent of</u> <u>WPEC Passed</u>
Bank Street Writer	3	10.0
Text Wizard	5	16.7
Letter Perfect	7	23.3
Speed Script	8	26.7
Atari Word Processor	9	30.0
Atari Writer	9	30.0
The Writer's Tool	9	30.0
Paper Clip	14	46.7
Atari Writer Plus	15	50.0
- - - - -		
Rewrite	30	100.0

Note. These results relate to the 30 word processing evaluation criterion (WPEC) listed in Table 1.

Table 1 which were supported by each of the word processors studied. Since all nine of the word processors in the sample have less than 75 percent of these capabilities, hypothesis two is accepted.

Hypothesis Three

It is possible to design and implement a word processing program for the Atari computer which has more of the capabilities necessary to meet the currently unsatisfied word processing needs of teachers, and more capabilities overall, than any of the popular word processors available for this computer.

The first requirement for acceptance.

Since all of the word processing capabilities determined to be needed by teachers (see Table 1) were successfully implemented in Rewrite (see Table 2), and since this is twice as many as were found in any of the word processors in the sample, the first requirement for the acceptance of hypothesis three has been demonstrated. A brief description of how these capabilities were implemented is given below.

Rewrite combines the simplicity of menu options with the efficiency of key commands. The beginner and infrequent user are provided with easy access to nearly all of the features through three menus. (See Appendix C, pp. 75-76 & 79-93.) Yet, the experienced user need not be hampered by the menu structure because key command alternatives are provided for most operations. (See

Appendix C, pp. 92-103.) Moreover, help files can be viewed without losing the text in memory. (See Appendix C, p. 80.)

One of the most unique characteristics of Rewrite is the accuracy of the display produced by the scrolling screen editor. While it does not always show an exact representation of the printed page, the knowledgeable user should seldom be surprised by what is produced. There are several design elements which work together to make this possible.

The same margin settings that are used to define the format of the printed page are used to display text on the screen. These settings can be defined globally (see Appendix C, pp. 86-87) or by commands contained within a document (see Appendix C, pp. 89 & 105). The latter method facilitates the accurate display of parts of a document which have different settings.

There are three horizontal settings (measured in characters from the left-most print position on the paper) which define the page layout: left margin, indent margin, and right margin. Line lengths effected by these settings can range from 2 to 213. The first line of each block of text is indented when the left margin is set to the left of the indent margin and outdented when the reverse is done. If desired, outdented blocks (as well as any items) can be automatically numbered when printed. (See Appendix C, p. 106.)

Documents can be displayed in a nonscrolling format to facilitate proofreading. Entry to and exit from the proofreading mode can be accomplished quickly with only three keystrokes. (See Appendix C, p. 87.)

The word wrap logic that is used to parse a document into lines on the printed page is also used by the screen editor. This assures an accurate display of line content. With few exceptions, the horizontal positioning of items is also accurate; this simplifies visual alignments of items into columns. Three other features aid visual alignments: selectable tab stops (see Appendix C, pp. 95-107), a horizontal print position scale (see Appendix C, p. 74), and a vertical guide bar which extends the height of the screen through the cursor (see Appendix C, pp. 88 & 94).

The false spaces produced by the word wrap process, as well as those produced by automatic indenting and outdenting, can be displayed as colored blocks to distinguish them from regular spaces (see Appendix C, pp. 88 & 94). This is useful when working with text as well as data files.

Rewrite also supports other features which aid with data file editing. The word wrap feature may be disabled for both displaying and printing (see Appendix C, pp. 88 & 94). Any of the 256 ASCII characters (including the return character) can be used in search and replace operations (see Appendix C, p. 88) and can be saved on a disk without alteration.

Single character macro printer codes can be defined (and repeatedly redefined throughout a document) to send any sequence of up to 32 control codes to the printer. This allows most printing features to be used by imbedding macros within the text. (See Appendix C, pp. 106-107.)

Macro printer codes also provide a method of producing characters which are not normally available on printers. For example, if a macro is defined to be "0" followed by the backspace command and "/", the character "Ø" will be printed whenever the macro is used.

One of the macro codes allows printer initialization codes to be sent to the printer automatically with each new page. (See Appendix C, pp. 106-107.) This insures that pages are printed as intended when the printer is turned off between pages and when only selected pages are printed.

There are also two macro printer codes that are invoked by a change in the video of the text. (See Appendix C, p. 106.) These allow such things as exponents, underlining, and/or bold print to be effected by inverse video text rather than with imbedded macro codes, which distort the horizontal positioning of text on the screen.

Large sequences of control codes can be sent to the printer by imbedding them within a document or by specifying within a document the name of a file which contains the desired codes. (See Appendix C, p. 108.) The latter method was used to provide over eighty different symbols used in mathematics by redefining the shapes of

MONEY		GEOMETRY & TRIG		CHAR	EXP	SUB
Cent symbol	¢	Pi	π	1	1	1
		Written "1"	ℓ	2	2	2
SET NOTATION		Parallel		3	3	3
Empty set	∅	Perpendicular	⊥	4	4	4
Union	U	Congruent	≅	5	5	5
Intersection	∩	Angle	∠	6	6	6
Left brace	{	Degree	°	7	7	7
Right brace	}	Triangle	Δ	8	8	8
Subset of	⊂	Parallelogram	▭	9	9	9
Element of	∈	Similar	~	0	0	0
		Circle	⊙	+	+	+
RELATIONAL		Identity	≡	-	-	-
Less or equal	≤	Left arrow	←	=	=	=
Greater or equal	≥	Right arrow	→	i	i	i
Approx. equal	≈	Segment bar	—	j	j	j
		Left arc	⌒	k	k	k
OPERATIONS		Right arc	⌢	n	n	n
Plus or minus	±			r	r	r
Multiply	•	GREEK LETTERS		x	x	x
Divide	÷	Sigma (Capital)	Σ	y	y	y
Composition	◦	Omega (Capital)	Ω	z	z	z
		Alpha	α	Infinity		∞
SQUARE ROOT		Beta	β	EXAMPLES:		
Top bar	—	Gamma	γ			
Top left corner	┌	Delta	δ	(x ₁ , y ₁)		
Vertical bar		Theta	θ			
Bottom left	└	Mu	μ			
		Nu	ν	log ₄ 10		
MISCELLANEOUS:		Pi	π			
Absolute value		Sigma	σ	Σ a _{i,j}		
Grid Cross	+	Tau	τ			
X-bar (Mean)	$\bar{x}$	Phi	φ	nC _r & nP _r		
Integral	∫	Chi	χ			
		Omega	ω	sin ⁻¹ (x)=θ		
EXAMPLES:						

A = {1,3}

B = {2,4}

C = {2}

AUC = {1,2,3}

A ∩ B = {} or ∅

C ⊂ B & 1 ∈ A

$$|x| = 5 \leftrightarrow x = -5 \text{ or } x = 5$$

$$|x| \leq 5 \leftrightarrow -5 \leq x \leq 5$$

$$|x| \geq 5 \leftrightarrow x \leq -5 \text{ or } x \geq 5$$

$$\pi \approx 3.1415926535897932$$

$$3 \cdot 4 = 12, 12 \div 4 = 3$$

$$(f \circ g)(x) = f[g(x)]$$

$$\ell \parallel m, m \parallel n, \text{ \& } \ell \perp n$$

$$\angle A \cong \angle B \text{ \& } m\angle A = 30^\circ$$

$$\text{In } \odot R, \triangle ABC \cong \triangle DEF$$

$$\square ABCD \text{ is a square}$$

$$\sin^2 \theta + \cos^2 \theta \equiv 1$$

$$\chi^2\text{-test, } \bar{x} = \text{mean}$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\sigma = \sqrt{\frac{\sum (x - \mu)^2}{n}}$$

Figure 2. Printer font of mathematics symbols.

characters on a dot matrix printer (see Figure 2). This method can also be used to include diagrams, produced by graphics software, within a text document.

The print control methods used by Rewrite make it compatible with virtually any printer. The compatibility problem with televisions having overscan was eliminated by allowing the the width of the screen used by Rewrite to be adjusted between 30 and 40 characters. (See Appendix C, p. 85.)

Effecting compatibility with all of the Atari 8-bit computers having at least 32K of memory involved several considerations. First, all interfacing with the various operating systems was done exclusively through those entry points which were the same for all of the models. Second, memory-efficient programming techniques were used extensively. Third, a software method was found to automatically disable the built-in BASIC of the XL and XE models. Fourth, precautions were taken to avoid conflicts between needed interrupt processing and the RAM bank selecting of the 130XE model computer. Finally, a method compatible with all of the operating systems was devised for increasing the automatic key repeat rate.

The second requirement for acceptance.

As shown in Table 3, Rewrite has over 20 percent more capabilities overall than any of the word processors sampled. Hypothesis three is therefore accepted.

Table 3

Overall Assessment of Word Processing Capabilities

Name of the Word Processor	Number of WPEC Passed	Percent of WPEC Passed
Bank Street Writer	39	26.0
Text Wizard	60	40.0
Atari Word Processor	61	40.7
Speed Script	69	46.0
Atari Writer	73	48.7
Letter Perfect	79	52.7
Atari Writer Plus	89	59.3
The Writer's Tool	96	64.0
Paper Clip	101	67.3

Rewrite	132	88.0

Note. These results relate to the 150 word processing evaluation criterion (WPEC) listed in Appendix B.

Chapter Six

Conclusions and Applications

Through this study, word processing capabilities needed by teachers were identified; nine existing word processors were evaluated and found lacking; and a new word processor, entitled Rewrite, was developed which has many superior features specifically designed for teachers. While many are expected to benefit from this study, Rewrite does have a number of limitations.

Features Not Implemented in Rewrite

Unnecessary duplication avoided.

Features which would have met six of the WPEC were not implemented because similar results could be obtained by the use of a combination of features that were implemented. The explanations follow.

1. The ability to conditionally designate where long words should be hyphenated (WPEC #40) is especially helpful when a screen editor cannot accurately display where lines will be parsed when the document is printed. Since the screen editor implemented in Rewrite gives an accurate display of parsing, this feature was not included.

2. Cursor movement by sentence (WPEC #46) was not directly supported because of a more general cursor movement operation that advances the cursor to the next of many special characters including the period, question mark, and exclamation point. (See Appendix C, p. 98.)

3. The block copy operation (WPEC #76) was not implemented because of the many retrievable delete operations that can be used to accomplish the same task. (See Appendix C, pp. 90-91 & 99.)

4. The purpose of notifying the user that a file requested to be saved already exists on the disk (WPEC #87) is to prevent accidental loss of a previously saved file with the same name. This same protection can be afforded by locking files that need such protection. (See Appendix C, p. 81.) Since wild card characters can be used to specify files, all the files on a disk can be locked or unlocked with a single command. Therefore, to avoid extra queries each time an existing file is edited and saved, this automatic warning feature was not implemented.

5. The ability to load and keep two files separate in memory (WPEC #96) was not included because of the advanced implementations of three other operations. The save operation allows virtually any portion of a document to be saved as a single file. (See Appendix C, p. 81.) The load operation allows virtually any type of merging of two files to be carried out. (See Appendix C, p. 82.) Finally, by marking the text with special characters, the cursor can be

moved from one portion of the text to another. (See Appendix C, p. 98.) These three operations allow two portions of the text in memory to be treated as separate files. The added convenience of having this done automatically was not considered worth the extra programming that would be required.

6. The ability to display the document on a scrolling screen in a noneditable form, exactly as it will appear on the printed page (WPEC #104), was not included because little would be gained over the powerful screen editor that was implemented.

Incompatible features.

No means was found of meeting four of the WPEC without inhibiting the function of important features already implemented. The conflicts which arose are described below.

1. If the text were formatted into lines when entered, it would be an easy programming task to display the line number of the page or document that the cursor is on (WPEC #13). This type of formatting, however, makes immediate adjustments to new margin settings and to newly inserted text, quite difficult. Since the latter capabilities were preferred, the cursor line number feature was not included. The cursor, however, can be moved up or down ten lines at a time through the document. This facilitates the quick counting of lines by the user. (See Appendix C, p. 97.)

2. It is very helpful to the user to have a marked block highlighted in some way so that there is no question about what text is included within the block (WPEC #75). The programming methods that were used to attempt to implement this feature either slowed the responsiveness of the keyboard to an unacceptable level or required that the block be of a limited size. The decision was therefore made not to highlight marked blocks. To compensate for the lack of this feature, the added ability of being able to move the cursor to either the beginning or end of a marked block was included. (See Appendix C, p. 98.)

3. The high speed processing methods used to handle imbedded print commands did not allow justification at the right margin (WPEC #130).

4. The memory clearing techniques used to permit multiple files to be printed together as a single document did not permit multiple copies of a document to be printed automatically (WPEC #136).

Complex features.

Eight of the WPEC were not met by Rewrite because they were beyond the ability of this researcher to implement within the time constraints of this study. They are (a) keyboard buffering (WPEC #32), (b) word counting (WPEC #83), (c) conditional page breaking (WPEC #112), (d) double column printing in one pass (WPEC #139), (e) support for double density disk drives (WPEC #147), (f) eighty column display (WPEC #148), (g) mail merge capability (WPEC #149),

(h) and spelling checking (WPEC #150)

Applications

The overall impact of this study is expected to be very positive. Some of the teachers at J. I. Case High School have already benefited directly by using Rewrite. It is expected that many other interested teachers in the Racine Unified School District will receive the same direct benefits as this program is made available to them. The extent of the direct impact this study will have on teachers in general is unknown, since only a small percentage of all school districts and educational institutions have Atari computer systems. There may be more widespread indirect benefits for all teachers that use word processors, if some of the features which are unique to Rewrite are implemented on other computer systems.

References

- Ahlstrom, S. J. & Moore, D. L. (1985). Paperclip [Computer word processing program, version 1.2]. Ontario Canada: Batteries Included.
- Atari personal computer system operating system listing. (1981, February). Sunnyvale, CA: Atari.
- Atari personal computer disk utilities listing. (1981, February). Sunnyvale, CA: Atari.
- Atari personal computer system operating system user's manual. (1980a). Sunnyvale, CA: Atari.
- Atari personal computer system hardware manual. (1980b). Sunnyvale, CA: Atari.
- Atari Staff. (1985). Atari Writer Plus [Computer word processing program & manual]. Sunnyvale, CA: Atari.
- Atari Staff. (1982). Atari Writer [Computer word processing program & manual]. Sunnyvale, CA: Atari.
- Atari Staff. (1981). Atari Word Processor [Computer word processing program & manual]. Sunnyvale, CA: Atari.
- Basovski, J. D. (1986, Winter). Atariwriter Plus and Paperclip [Comparative software review]. Atari Explorer, 16-20.
- Brannon, C. (1985a). Speed Script: The word processor for Atari computers [Computer word processor manual & source code]. Greensboro, NC: Compute! Publications.
- Brannon, C. (1985b). Speed Script 3.0: All machine language word processor for Atari [Software overview]. Compute!, 7(5), 103-116.
- Brannon, C. (1985c). Speed Script [Computer word processing program, version 3.0]. Greensboro, NC: Compute! Publications.
- Card, O. S. (1984). Reading the keyboard codes. In Compute!'s third book of Atari (pp. 4-21). Greenboro, NC: Compute! Publications. Inc.
- Chadwick, I. (1983). Mapping the Atari. Greenboro, NC: Compute! Publications. Inc.

- Ciraolo, M. & Friedland, N. (1985, May). Paperclip: Start-of-the-art Atari word processing! [Software review]. Antic, 4(1), 24-26.
- Ertl, R. D. (1987). Rewrite [Computer word processing program & assembled source code listing, version 2.4]. Kenosha WI: Carthage College Computer Center
- Harding, S. (1985, May). Paperclip [Computer word processor manual]. Ontario Canada: Batteries Included.
- Kushner, R. (1983, April/May). The new Atari Writer word processor [Software review]. Analog Computing, (11), 107-111.
- Lawrow, S. D. (1982). MAC/65 [Computer macro assembler and manual]. San Jose, CA: Optimized Systems Software.
- Leventhal, L. A., Saville, W. (1982). 6502 assembly language subroutines. Berkeley, CA: OSBOURNE/McGraw-Hill.
- Leyenberger, A. (1985, Summer). Throw out the white-out: Wordprocessing and the Atari computer [Comparative software review]. Atari Explorer, 34-38.
- Leyenberger, A. (1984, August). Cut & Paste [Software review]. Analog Computing, (21), 71-72.
- LJK Staff. (1981). Letter Perfect [Computer word processing program]. St. Louis, MO: LJK Enterprises.
- Lyons, L. (1984, Winter). Professional Atari Writer [Software review]. Atari Connection, 31-35.
- OSS Staff. (1984). The Writer's Tool [Computer word processing program & manual, version 2.2]. San Jose, CA: Optimized Systems Software.
- Riggs, R. L. (1985, October). The Writer's Tool for Atari [Software review]. Compute!, 7(10), 74-76.
- Robinson, W. V. (1981). Text Wizard [Computer word processing program & manual, version 1.3]. Chatsworth, CA: DataSoft.
- Roquemoire, S. (1986, April). Atariwriter Plus [Software review]. Antic, 4(12), 81-84.
- Smith, F. E., Kusmiak, G., & Olson, C. (1982). Bank Street Writer [Computer word processing program]. San Rafael, CA: Broderbund Software.

Wilkinson, B. (1983). The Atari BASIC source book.
Greenboro, NC: Compute! Publications. Inc.

Wilkinson, B. (1982). Inside Atari DOS. Greenboro, NC:
Compute! Publications. Inc.

Zaks, R. (1980). Programming the 6502 (3rd ed.).
Berkeley, CA: Sybex.

Appendix A

Glossary of Computer Terms

ASCII. ASCII is an acronym for American Standard Code for Information Interchange. There are 256 standard characters included in this code.

Assembler. An assembler is a computer program which converts assembly language source code into a form (machine language) that the computer can execute directly.

Bank selecting. Bank selecting is the process of designating one of several possible blocks of memory to be accessed through a specific range of memory addresses.

Bit. A bit is a binary digit. When prefixed with a number, the number describes how many bits a computer can process at one time. Such numbers are usually multiples of eight.

Configure. To configure is to adjust or adapt a software package (e.g. a word processor) so that it will accommodate the capabilities of the peripheral equipment (e.g. a printer) with which it is used.

Cursor. A cursor is a marker on a screen display that shows where the next character typed will appear.

Data file. A data file is a collection of related information, accessible by a computer, that is treated as a unit for storage and retrieval purposes.

Delimiter. A delimiter is a special character that is

used before and/or after a portion of text to identify it in some way.

Disable. To disable is to prevent a signal from being acted on or a portion of memory from being used.

Dot matrix printer. A dot matrix printer is a printer that produces characters formed by an arrangement of dots.

Edit. To edit is to alter the arrangement and/or the content of a body of text or other data.

Field. A field is a collection of characters that constitute a single logical unit of data.

Footer. A footer is a special portion of text that is specified to be printed below the main text on one or more pages of a document.

Header. A header is a special portion of text that is specified to be printed above the main text on one or more pages of a document.

Interrupt. An interrupt is a signal which temporarily suspends the computer's normal sequence of operations and transfers control to a special routine.

Macro. A macro is a single entity that can be designated to have the same meaning or effect as one or more other such entities.

Operating system. An operating system is a computer program (usually stored in ROM) that controls the overall activity of a computer system.

Outdent. To outdent is to begin a line (or the first line of a block) of text to the left of the left margin.

Overscan. Overscan is the condition that exists when the intended screen display is not contained completely within the physical boundaries of the screen.

Print position. The print position is the location of a character within a printed line. It is specified by the number of characters from the left-most character that can be printed.

RAM. RAM is an acronym for Random Access Memory, which is memory that can be both read and altered.

Record. A record is a collection of related fields.

ROM. ROM is an acronym for Read Only Memory, which is memory that can be read but not altered.

Screen editor. A screen editor is a specialized screen display that facilitates an editing process.

Scroll. To scroll is to display text on a screen in a way that gives the illusion that the screen is a moving viewing window onto a larger document. The term usually implies that horizontal as well as the more common vertical "movement" of this viewing window is possible.

Source code. Source code is a computer program written in a programming language other than machine language.

Subroutine. A subprogram that can be executed from more than one place in the main program.

User. A user is a person who is using a computer program and/or system.

Wild card character. A wild card character is a

specialized character that is considered to be a match for any other character with which it is compared.

Word wrap. Word wrap is the process of parsing a body of text into lines without splitting words. Some variations of this process also permit parsing at hyphenation points.

Appendix B

Word Processor Evaluation: Criteria and Results

This appendix contains the list of word processor evaluation criteria (WPEC) that were used to evaluate the word processors. The results of these evaluations are shown in the matrix contained on the next four pages. Within this matrix, "Y" means "yes", "N" means "no", and "-" means "does not apply because the more general capability, to which this criterion relates, is not supported". The following codes were used at the top of each of these pages to identify the word processors:

<u>Code</u>	<u>Word Processor</u>
A	The Atari Word Processor
B	Text Wizard
C	Letter Perfect
D	Atari Writer
E	Bank Street Writer
F	The Writer's Tool
G	Paperclip
H	Atari Writer Plus
I	Speed Script
- - - - -	
Z	Rewrite

Itemized descriptions of each criterion conclude this appendix.

Word Processor Code:	A	B	C	D	E	F	G	H	I	Z
Overwrite mode										
25-Overwrite mode provided	N	Y	Y	N	N	Y	Y	Y	Y	Y
26-Back space opposite of space	-	Y	N	-	-	Y	N	Y	N	Y
27-Move over text with tab key	-	Y	Y	-	-	Y	Y	Y	N	Y
28-Insert text if on return char	N	N	N	N	N	N	Y	N	Y	Y
Text entry and conversion										
29-Insert large block of spaces	N	N	N	N	N	Y	N	N	Y	Y
30-Remove spaces to next char.	N	N	N	N	N	Y	N	N	Y	Y
31-All ASCII can be entered	Y	Y	N	N	N	Y	N	N	N	Y
32-Keyboard buffered	Y	Y	Y	N	N	N	Y	N	N	N
33-Fast automatic key repeat	N	N	N	N	N	N	Y	N	N	Y
34-Convert case of entered text	Y	N	N	Y	N	Y	Y	Y	N	Y
35-Convert video of entered text	Y	N	N	Y	N	N	N	Y	N	Y
36-Definable macro key	N	N	N	N	N	N	Y	N	N	Y
Word wrap parsing										
37-Word wrap during text entry	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
38-Immediate during text entry	N	Y	Y	Y	N	Y	Y	Y	Y	Y
39-Can be turned off	N	N	N	N	N	Y	N	N	N	Y
40-Also on soft hyphens	N	N	N	N	N	Y	Y	N	N	N
41-Not on hard spaces	N	N	N	N	N	Y	Y	N	N	Y
Cursor movement										
42-By character	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
43-By line	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
44-By multiple lines	N	Y	Y	Y	Y	Y	Y	Y	N	Y
45-By word	N	N	Y	N	N	Y	Y	Y	Y	Y
46-By sentence	N	N	N	N	N	N	N	N	Y	N
47-By paragraph	N	N	Y	N	N	N	N	N	Y	Y
48-With tab key	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
49-To selectable tab stops	Y	N	Y	Y	N	N	Y	Y	N	Y
50-To beginning of line	Y	N	Y	Y	N	Y	Y	Y	N	Y
51-To end of line	Y	N	Y	Y	N	Y	Y	Y	N	Y
52-To top of screen	N	N	Y	N	N	N	N	N	Y	Y
53-To top of document	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
54-To bottom of document	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
55-To special markers	N	N	N	N	N	N	Y	N	N	Y
Erase text in memory										
56-Erase all text	N	Y	Y	N	Y	N	Y	N	Y	Y
57-Erase text above cursor	N	N	Y	N	N	Y	Y	N	N	Y
58-Erase text below cursor	N	Y	Y	Y	N	Y	Y	Y	N	Y
59-Erase any size block of text	N	N	Y	N	N	Y	Y	N	N	Y
Delete & retrieve elsewhere										
60-Retrieveable deletions	N	N	Y	Y	N	Y	Y	Y	Y	Y
61-Multiple deletions	-	-	Y	N	-	Y	N	Y	Y	Y
62-Multiple retrievals	-	-	Y	Y	-	Y	Y	Y	Y	Y
63-Maximum size at least 1K	-	-	Y	N	-	N	N	Y	Y	Y

Word Processor Code:

A	B	C	D	E	F	G	H	I	Z
---	---	---	---	---	---	---	---	---	---

Search & replace

64-Conditional	Y	Y	Y	Y	Y	Y	Y	Y	Y
65-Repeatable without resetting	N	Y	N	N	N	Y	Y	Y	Y
66-Global	Y	N	Y	Y	N	Y	Y	Y	Y
67-Next search past last replace	Y	Y	Y	Y	Y	Y	N	Y	Y
68-Wild card search character	N	N	N	N	N	Y	N	N	Y
69-Wild card replace character	N	N	N	N	N	N	N	N	Y
70-Search for any control char.	Y	Y	N	N	N	N	N	N	Y
71-Search for escape character	N	Y	N	N	N	Y	N	N	Y
72-Search for return character	N	N	N	N	N	Y	N	N	Y
73-Search thru 12K is 2 seconds	N	Y	N	Y	N	Y	Y	Y	Y

Block operations

74-Block operations available	Y	Y	Y	Y	Y	Y	Y	Y	N
75-Block highlighted	N	N	N	N	Y	Y	Y	Y	-
76-Block copy	Y	Y	Y	Y	N	Y	Y	Y	-
77-Block move	Y	Y	Y	N	Y	N	Y	N	-
78-Block delete	Y	Y	Y	Y	Y	Y	Y	N	-
79-Block size can be 2K	Y	N	Y	N	N	Y	N	N	-

Text Memory

80-Text memory at least 20K	N	Y	Y	Y	N	Y	N	N	Y
81-Protection from overfilling	E	N	Y	Y	Y	Y	Y	Y	Y
82-Free memory notification	Y	Y	Y	Y	N	Y	Y	Y	Y
83-Count total words used	N	Y	Y	N	N	Y	Y	Y	N

Disk operations

84-Device prefix is optional	-	N	Y	Y	-	Y	Y	Y	N
85-Filename extenders allowed	N	Y	N	Y	Y	Y	Y	Y	Y
86-Automatic save or reminder	N	N	N	N	N	N	E	N	N
87-Save warning if file exists	Y	N	N	Y	N	N	N	Y	N
88-Save any part of a document	N	Y	N	N	Y	Y	Y	N	N
89-Save only entered data	N	Y	Y	Y	Y	Y	Y	Y	Y
90-All ASCII preserved	N	Y	Y	Y	N	Y	Y	Y	N
91-Previous file as save default	Y	N	Y	Y	N	Y	Y	Y	N
92-Previous file as load default	Y	N	N	N	N	N	N	N	N
93-Partial overwrite load	N	Y	N	Y	N	Y	E	N	Y
94-Cursor anywhere & append load	N	N	Y	N	N	N	Y	N	N
95-Insert load	N	N	N	N	Y	N	N	Y	N
96-Two files in memory	N	N	N	N	N	N	Y	N	N
97-View file without losing text	N	N	N	N	N	N	Y	N	N
98-Disk directory	Y	Y	Y	Y	Y	Y	Y	Y	Y
99-Delete files	Y	Y	Y	Y	Y	Y	Y	Y	Y
100-Initialize a diskette	N	N	Y	Y	Y	Y	Y	Y	Y
101-Rename files	N	N	N	N	Y	N	Y	N	N
102-Lock & unlock files	N	N	Y	N	N	N	Y	N	Y

Print preview

103-Preview document on screen	Y	Y	Y	Y	N	Y	Y	Y	Y
104-Preview full width of page	Y	N	N	Y	-	N	Y	Y	N
105-Works without printer	Y	Y	Y	Y	-	N	Y	Y	Y

Word Processor Code:

A B C D E F G H I Z

Print control

106-Adjustable external formats	Y	N	N	N	Y	Y	N	Y	N	Y
107-Formats adjustable in text	N	Y	Y	Y	N	Y	Y	Y	Y	Y
108-Select margins	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
109-Select line spacing	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
110-Select paragraph spacing	Y	N	N	Y	N	N	N	Y	N	Y
111-Forced page breaks	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
112-Conditional page breaks	N	N	N	N	N	Y	N	N	N	N
113-Indent marked line	N	N	N	Y	N	Y	Y	Y	N	Y
114-Automatic indenting	N	N	N	N	N	N	N	N	N	Y
115-Outdent marked line	N	N	N	N	N	Y	Y	N	Y	Y
116-Automatic outdenting	N	N	N	N	N	N	N	N	N	Y
117-Centering	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
118-Above with partial lines	N	N	N	N	N	Y	Y	N	N	Y
119-Block at right margin	Y	Y	N	Y	N	Y	Y	Y	Y	Y
120-Above with partial lines	N	N	N	N	N	Y	Y	N	N	Y
121-Underlining	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
122-Above without delimiters	Y	N	N	Y	N	N	N	Y	N	Y
123-Superscripts & subscripts	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
124-Above without delimiters	N	N	N	N	N	N	N	N	N	Y
125-Selectable typeface	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
126-Headers & footers	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
127-Adjustable position of above	N	N	Y	N	N	Y	Y	N	N	Y
128-Page numbering	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
129-Item numbering	N	N	N	N	N	N	N	Y	N	Y
130-Justification	Y	Y	Y	Y	N	Y	Y	Y	N	N
131-Double space handled properly	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
132-Single sheet printing	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
133-Initialize printer each page	Y	N	N	N	N	N	N	N	N	Y
134-Select pages to print	Y	N	Y	Y	N	Y	Y	Y	Y	Y
135-Odd or even pages only	N	N	N	N	N	N	N	N	N	Y
136-Multiple copies	N	N	Y	Y	N	Y	Y	Y	N	N
137-Preset files to link	N	Y	N	Y	N	Y	Y	Y	Y	Y
138-Batch file for printing files	Y	N	N	N	N	N	Y	N	N	Y
139-Double column in one pass	Y	N	N	N	N	N	E	Y	N	N
140-Skip remarks	N	N	N	N	N	Y	N	N	Y	Y
141-Verbatim codes to printer	Y	N	Y	Y	N	Y	N	Y	N	Y
142-Include printer code files	N	N	N	N	N	Y	Y	N	N	Y

Miscellaneous

143-Error reporting	Y	N	Y	Y	Y	Y	Y	Y	Y	Y
144-Memory protected from RESET	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
145-New initial defaults possible	N	N	N	N	N	N	Y	N	N	Y
146-Adaptable to other DOS	N	N	N	Y	N	Y	N	N	Y	Y
147-Supports double density disks	N	N	Y	N	N	Y	Y	N	N	N
148-80 column version available	N	N	Y	N	N	N	Y	N	N	N
149-Mail merge capability	N	N	Y	N	N	Y	Y	Y	N	N
150-Spelling checker included	N	N	Y	N	N	Y	N	Y	N	N

Descriptions of the Word Processing Evaluation Criteria Used

Compatibility

- 1-Compatible with most printers: A means is provided to use the special printing features with any capable printer without additional cost.
- 2-800, XL, & XE compatible: The program will work properly on the Atari 800, XL, and XE model computers without the use of an operating system translator.
- 3-Disables XL & XE BASIC: The word processor automatically disables BASIC on these Atari computer models without having to hold the option key down while turning the computer on.
- 4-Atari DOS compatible: The word processor can produce text files that are compatible with Atari DOS 2.0S or 2.5.
- 5-32K of RAM is sufficient: The word processor will function properly with only 32,768 bytes of random access memory.

Command System

- 6-Menus with one or two levels: Menus with one or two levels are provided for carrying out operations easily.
- 7-Keyboard Commands: Some operations can be carried quickly by simply pressing one key or a combination of keys.

Screen Display

- 8-Adjustable width: The maximum number of characters that will be displayed on each screen line is adjustable to compensate for televisions or monitors with overscan.
- 9-Width is at least 40 characters: The maximum number of characters that can be displayed on each screen line is at least 40.
- 10-Height at least 20 lines: The maximum number of lines of text that are displayed on the screen at one time is at least 20.
- 11-Improved legibility: The word processor makes use of a special text mode that makes the characters more legible than those normally used by the computer.

- 12-Selectable color: The color of the screen display is selectable.

Editing Display

- 13-Line of page or document: The number of the line on the page or in the document that the cursor is on is indicated.
- 14-Screen column indicator: The screen column position of the cursor is indicated.
- 15-Other alignment aid provided: Some visual means is provided for positioning characters that are on different lines so that one is directly beneath the other.
- 16-False spaces markable: The spaces that are not truly part of the document (such as those caused by word wrap, automatic indenting, or automatic outdenting) can be distinguished from the spaces that were actually typed.

Scrolling editing display

- 17-Scrolling editor available: The text editor can be set to scroll across lines of text that are longer than the maximum screen width.
- 18-Accurate line content: The screen editor can be set so that the text that appears on any line is the text that will be printed on that line.
- 19-Uses the text margin settings: The screen editor is adjusted using the same settings that are used to adjust the text on the printed page.
- 20-Maximum length at least 200: The maximum number of characters that can be treated as one line by the screen editor is at least 200.
- 21-Quick switch to proofread: It is possible, with any size of document, to switch between the scrolling screen editor and an editor in which each line can be displayed completely on the screen in less than five seconds.

Insert mode

- 22-Insert mode provided: A text entry mode is provided in which new text is inserted into the existing text as it is entered.

- 23-Back space opposite of space: In insert mode, the back space key and the space bar perform opposite operations. This means that the back space key will remove the character that is to the left of the cursor rather than simply replace it with a space.
- 24-Insert spaces with tab key: When the tab key is pressed in insert mode, spaces are inserted as the cursor moves to the next tab stop.

Overwrite mode

- 25-Overwrite mode provided: A text entry mode is provided in which new text replaces existing text as it is entered.
- 26-Back space opposite of space: In overwrite mode, the back space key and the space bar perform opposite operations. This means that the back space key will replace the character that is to the left of the cursor with a space, rather than remove it from the text memory.
- 27-Move over text with tab key: When the tab key is pressed in overwrite mode, the text in the text memory is not altered as the cursor moves to the next tab stop.
- 28-Insert text if on return character: When the cursor is positioned on a return character, the next character entered is inserted in front of the return character rather than in place of the return character.

Text entry and conversion

- 29-Insert large block of spaces: There is an operation that will enter more than one screen line of spaces at the cursor position, so that new text can be entered quickly in overwrite mode.
- 30-Remove spaces to next character: There is an operation that will remove the contiguous block of spaces between the cursor position and the next nonspace from the text memory.
- 31-All ASCII can be entered: There is a way to enter every ASCII character from the keyboard.
- 32-Keyboard buffered: Keys that are pressed too quickly for the word processor to process immediately are not ignored, but processed as soon after they are pressed as possible.

- 33-Fast automatic key repeat: The keyboard keys can be set to automatically repeat more quickly than is normal (which is 10 characters each second).
- 34-Convert case of entered text: There is an operation that allows previously entered text to be converted to upper case or lower case without actually typing the desired letters.
- 35-Convert video of entered text: There is an operation that allows previously entered text to be converted to regular video or inverse video without actually typing the desired letters.
- 36-Definable macro key: It is possible to define a key command to produce the same effect as pressing a series of keys.

Word wrap parsing

- 37-Word wrap during text entry: No line ends with an inappropriate number of spaces or broken words within five seconds after new text is entered.
- 38-Immediate during text entry: The word wrap process described above is carried out with each key stroke as new text is entered.
- 39-Can be turned off: The word wrap process can be turned off so that all the lines end at the same column.
- 40-Also on soft hyphens: A special soft hyphen character can be entered in the middle of a word so that if the whole word will not fit on the end of a line, but the part up to the soft hyphen will, the line will be ended at the soft hyphen character, and a real hyphen will be printed there. If such a word is in the middle of a line, the soft hyphen character is ignored.
- 41-Not on hard spaces: A special hard space character, which is printed as a regular space, can be used to keep the adjacent items on the same line.

Cursor movement

- 42-By character: The cursor can be moved to the previous or to the next character in the document with a single command.

- 43-By line: The cursor can be moved to the same position on the previous line or on the next line in the document with a single command.
- 44-By multiple lines: The cursor can be moved backward or forward several lines in the document with a single command.
- 45-By word: The cursor can be moved to the next word in the document with a single command.
- 46-By sentence: The cursor can be moved to the beginning of the next sentence in the document with a single command.
- 47-By paragraph: The cursor can be moved to the beginning of the next paragraph in the document with a single command.
- 48-With tab key: The cursor is moved in some way by the tab key.
- 49-To selectable tab stops: The cursor can be moved to the next user selectable tab stop.
- 50-To beginning of line: The cursor can be moved to the beginning of a line with a single command.
- 51-To end of line: The cursor can be moved to the end of a line with a single command.
- 52-To top of screen: The cursor can be moved to the top of the existing screen display with a single command.
- 53-To top of document: The cursor can be moved to the top of the document in the text memory with a single command.
- 54-To bottom of document: The cursor can be moved to the bottom of the document in the text memory with a single command.
- 55-To special markers: The cursor can be advanced through the text to the next of several specially marked positions in the text.

Erase text in memory

- 56-Erase all text: The entire text memory can be erased with a single operation.
- 57-Erase text above cursor: The text from the beginning of the document to the cursor position can be removed from the text memory.

- 58-Erase text below cursor: The text from the cursor to the end of the document can be removed from the text memory.
- 59-Erase any size block of text: Any portion of text in the text memory can be removed from the text memory.

Delete & retrieve elsewhere

- 60-Retrieveable deletions: Deletions of text can be made and retrieved elsewhere within the document.
- 61-Multiple deletions: Two or more consecutive deletions can be completely retrieved later.
- 62-Multiple retrievals: More than one copy of previously deleted text can be retrieved.
- 63-Maximum size at least 1K: The maximum number of characters that can be deleted at one time is at least 1024.

Search & replace

- 64-Conditional: Searches for user-defined text can be executed and conditionally replaced within a document.
- 65-Repeatable without resetting: The search and replace operations can be repeated at any time without resetting the search or replacement specifications.
- 66-Global: It is possible to replace every portion of text that matches a search specification with the text of the replacement specification with a single command.
- 67-Next search past last replace: When a global search and replace is executed, each subsequent search begins with the next character after the last replacement that was made. (Without this, an unending memory filling operation is carried out when the search specification is contained within the replacement specification.)
- 68-Wild card search character: A special character may be included in the search specification that will be considered a match with any character during a search operation. This feature allows similar items in the text to be replaced with the same item.

- 69-Wild card replace character: A special character may be included in the replacement specification that will cause the corresponding character in a document to remain unaltered during a replacement operation. This feature allows similar items in a document to be altered in similar ways without altering the parts that must be preserved.
- 70-Search for any control character: Any control character may be included in the search specification.
- 71-Search for escape character: It is possible to include the escape character within the search specification.
- 72-Search for return character: It is possible to include the return character within the search specification.
- 73-Search thru 12K in 2 seconds: The maximum time required to locate and display a match for the search specification that appears at the end of 12,288 characters of text is 2 seconds.

Block operations

- 74-Block operations available: It is possible to mark a block of text and carry out an operation on that text.
- 75-Block highlighted: When a block of text is marked, the text within is highlighted.
- 76-Block copy: It is possible to make a copy of a marked block anywhere within the text memory.
- 77-Block move: It is possible to move a marked block from one part of the text memory to another.
- 78-Block delete: It is possible to remove a marked block from the text memory.
- 79-Block size can be 2K: The maximum number of characters that can be included within a marked block is at least 2048.

Document Size

- 80-Text memory at least 20K: The maximum number of characters that may be contained in the text memory at one time is at least 20,480.

- 81-Protection from overfilling: The program continues to work properly after attempting to put more text into the text memory than it is able to hold.
- 82-Free memory notification: A means is provided of monitoring the amount of free space in the text memory.
- 83-Count total words used: The word processor can display the number of words contained in the text memory.

Disk operations

- 84-Device prefix is optional: It is acceptable to either include or to omit the disk device name when specifying a disk file.
- 85-Filename extenders allowed: It is possible to include a filename extender when specifying a disk file.
- 86-Automatic save or reminder: The program, periodically, either automatically saves the text that is in the text memory or reminds the user to save it.
- 87-Save warning if file exists: When a save operation is attempted, a warning is first given if a file with the specified name already exists. The user is then queried as to whether the save operation is to be continued.
- 88-Save any part of a document: It is possible to save any designated part of the document which is in the text memory.
- 89-Save only entered data: It is possible to save the actual information that was entered without any extra data generated by the word processor.
- 90-All ASCII preserved: Save and load operations do not alter any of the 256 ASCII characters.
- 91-Previous filename as save default: When the name of a file is to be entered to save a document, the name used for the most recent save or load operation is provided as a default.
- 92-Previous filename as load default: When the name of a file is to be entered to load a document, the name used for the most recent save or load operation is provided as a default.

- 93-Partial overwrite load: When a file is loaded into the text memory, it can be designated to overwrite the text that is currently in the text memory, beginning at the cursor position.
- 94-Cursor anywhere & append load: When a file is loaded into the text memory, it can be designated to be appended to the text that is currently in the text memory, regardless of where the cursor is positioned.
- 95-Insert load: When a file is loaded into the text memory it can be designated to be inserted into the text that is already in the text memory at the current cursor position.
- 96-Two files in memory: Two parts of the text memory are treated as separate files during editing and during save and load operations.
- 97-View file without losing text: It is possible to view the contents of a disk file without altering the text that is in the text memory.
- 98-Disk directory: It is possible to view the names of the files on a disk.
- 99-Delete files: It is possible to delete any specified file(s) from a disk.
- 100-Initialize a diskette: It is possible to format a diskette.
- 101-Rename files: It is possible to assign a new name to an existing disk file.
- 102-Lock & unlock files: It is possible protect and unprotect disk files from alteration and erasure.

Print preview

- 103-Preview document on screen: A means of previewing a document on the screen, in the form that it will be printed, is available.
- 104-Preview full width of page: The document preview feature displays the full width of the document, just as it will appear when printed.
- 105-Works without printer: The document preview feature works without having a printer or printer interface as part of the computer system.

Print control

- 106-Adjustable external formats: There are page formats, including margin settings, that can be set without imbedding them in the text of the document.
- 107-Formats adjustable in text: The page formats can be adjusted repeatedly from within the document.
- 108-Select margins: Margins can be selected.
- 109-Select line spacing: The number of lines that the paper will be advanced after each printed line is selectable.
- 110-Select paragraph spacing: The number of lines that the paper will be advanced after each printed paragraph is selectable.
- 111-Forced page breaks: It is possible to end a page sooner than it would normally be ended.
- 112-Conditional page breaks: It is possible to specify that a specific group of lines be printed on the same page. If a group of lines so specified will not fit on the page that is being printed, they will all be printed on the next page.
- 113-Indent marked line: It is possible to designate individual lines that are to be indented. This allows paragraphs to be indented without actually typing the indent spaces.
- 114-Automatic indenting: An automatic feature is provided for indenting the first line of a block of text without marking such lines and which is accurately displayed by the screen editor.
- 115-Outdent marked line: It is possible to designate individual lines that are to begin left of the left margin position that is usually used. This permits characters such as numbers to be printed to the left of an otherwise left justified block of text, as in this listing.
- 116-Automatic outdenting: A feature is provided for automatically outdenting the first line of a block of text without marking such lines and which is accurately displayed by the screen editor. This permits characters such as numbers to be automatically printed to the left of an otherwise left justified block of text, as in this listing.
- 117-Centering: The word processor can center any line of text on the printed page.

- 118-Centering with partial lines: The word processor can center a portion of any line of text on the printed page.
- 119-Block at right margin: The word processor can print any line of text flush with the right margin.
- 120-Block at right margin with partial lines: The word processor can print the last portion of any line of text flush with the right margin.
- 121-Underlining: It is possible to cause text to be underlined on the printed page.
- 122-Underlining without delimiters: It is possible to cause text to be underlined on the printed page without using imbedded codes to delimit the text that is to be underlined.
- 123-Superscripts & subscripts: It is possible to print superscripts and subscripts.
- 124-Superscripts & subscripts without delimiters: It is possible to print superscripts and subscripts without using imbedded codes to delimit them.
- 125-Selectable typeface: It is possible to select the typeface(s) that will be used.
- 126-Headers & footers: Headers and footers can be defined to appear on each page, or selected pages, of the document when it is printed.
- 127-Selectable position of above: The exact line on which a header or footer will appear on the printed page is selectable.
- 128-Page numbering: Pages can be numbered automatically when the document is printed.
- 129-Item numbering: Items within the text of a document can be numbered automatically when the document is printed.
- 130-Justification: It is possible to have both the beginning and end of each full printed line appear flush with the left and right margins, respectively.
- 131-Double space handled properly: When a document is printed, no line begins with a single space because of two consecutive spaces used in the text, such as would be present after a colon or period.

- 132-Single sheet printing: It is possible to cause printing to be halted temporarily at the end of each page, which allows a new sheet of paper to be put into the printer before the next page is printed.
- 133-Initialize printer each page: It is possible to set printer initialization codes that will be sent to the printer at the beginning of each page.
- 134-Select pages to print: It is possible to print selected pages of a document.
- 135-Odd or even pages only: It is possible to print only the odd numbered pages or only the even numbered pages of a document. This allows a document to be printed on both sides of the paper with different page formats on each side.
- 136-Multiple copies: It is possible to specify that more than one copy of a the document be printed.
- 137-Preset files to link: When the text of a document is contained in more than one file, it is possible, for printing purposes, to designate the name of the next file of that document at the end of each file.
- 138-Batch file for printing files: It is possible to designate within a single file all of the files that are to be printed as part of a document.
- 139-Double column in one pass: It is possible to print text in two columns on a page without having the same page of a document pass through the printer more than once.
- 140-Skip remarks: It is possible to imbed remarks within a document that will not be printed.
- 141-Verbatim codes to printer: It is possible to designate any of the 256 ASCII codes to be sent to the printer from within the text of a document.
- 142-Include printer code files: It is possible to designate printer code files (e.g. for graphics or redefining character shapes) to be sent to the printer from within a document.

Miscellaneous

- 143-Error reporting: When an error occurs, it is reported to the user.

- 144-Memory protected from RESET: The text memory is not altered or cleared when the SYSTEM RESET button is pressed.
- 145-New initial defaults possible: New initial default settings can be established by the user. The initial default settings are the ones that are active immediately after the word processor is loaded into the computer.
- 146-Adaptable to other DOS: The word processing program can be adapted for use with other disk operating systems.
- 147-Supports double density disks: The word processing program supports the use of double density disk drives.
- 148-80 column version available: There is a version of this program available that will display 80 columns of text on the screen. Additional hardware may be required, however.
- 149-Mail merge capability: The word processor has the ability to replace parts of one document with specific information obtained from another file while the document is being printed.
- 150-Spelling checker included: The word processing program comes with a program or feature that allows the spelling of the words in a document to be checked.

Appendix C

Rewrite Reference Manual

This appendix contains the reference manual for the Rewrite word processor which was developed through this study. The table of contents for this manual, contains page numbers of the manual itself which are located in the lower right corner of each page. Pages referenced in the text of this paper refer to the paper's page numbers, which are located in the upper right corner.

TABLE OF CONTENTS

INTRODUCTION.....	1
SUMMARY OF THE FEATURES.....	2
The Screen Display.....	2
Text Entry and Editing.....	3
Print Control.....	4
Disk Operations.....	5
GETTING STARTED.....	6
Equipment You Will Need.....	6
Making a Back Up Copy of Your Program Diskette...	6
Loading the Program.....	6
ESSENTIAL INFORMATION.....	6
The Screen Display.....	6
Symbols for Keys and Key Combinations.....	7
The Three Main Menus.....	8
Responding to Prompts.....	10
Device Names and File Specifications.....	10
Using the BREAK Key.....	11
THE OPTION MENU.....	12
Disk Directory.....	13
Format Disk.....	13
View/Verify File.....	13
Erase, Write-Protect & Unprotect.....	14
Rename File.....	14
Save File.....	14
Load File.....	15
Print Document.....	16
THE SELECT MENU.....	17
Free Memory.....	17
Block Beginning & End.....	17
Display Color & Width.....	18
Keystrokes Before Save.....	18
Linefeed with Return.....	19
Page Format Parameters.....	19
Search & Replacement Specifications.....	21
Toggle Modes.....	21
THE START MENU.....	23
Block Operations.....	23
Delete Text to Buffer.....	23
Erase Text.....	24
Insert Operations.....	25
Search and/or Replace.....	26
Text Conversion.....	26

KEY COMMANDS.....	27
Screen Display Settings.....	27
Keyboard Settings.....	28
Cursor Movement.....	29
Other Methods of Moving the Cursor.....	31
Cut and Paste Operations.....	32
Erase Operations.....	32
Inserting Spaces at the Cursor.....	33
Search and Replacement.....	34
The Macro Key Command.....	35
Special Input Line Key Commands.....	35
IMBEDDED PRINT CONTROL COMMANDS.....	37
Horizontal Alignment Commands.....	37
Page Control Commands.....	37
Defining Headers and Footers.....	38
Page and Item Number Commands.....	38
Macro Printer Codes.....	39
Print Mode Toggles.....	40
Miscellaneous Print Control Commands.....	41
SUGGESTIONS AND HINTS.....	42
Methods of Inserting Text Quickly.....	42
Establishing a Pseudo Printer Driver.....	42
Printing Very Large Documents.....	43
REFERENCE SHEETS	

INTRODUCTION

The Rewrite Word Processor is a powerful word processing program for the original Atari home computer. It is mainly a menu driven program with many key command short cuts to the menu operations. The menus make it very easy for the beginner to learn how to use this word processor. The experienced and frequent user will find the key command short cuts to be more efficient. Many of these key commands involve key combinations that normally have no function on the Atari computer. Great care was taken to match the various operations with the key combinations in a meaningful way. This makes their use seem quite natural with a little practice. The experienced Atari computer user will be glad to find the majority of the normal keyboard functions preserved.

The following section is a brief summary of most of the features, capabilities and unique aspects of the Rewrite Word Processor. It basically describes what the program can do, without explaining how. The inexperienced user should not expect to understand all of the features listed immediately.

SUMMARY OF THE FEATURES

The Screen Display

- * The screen can be set to any of sixteen colors with either a dark background and light text or vice versa.
- * Extra tall characters are used to provide an extremely readable display.
- * The screen display acts like a window onto the document in the computer's memory. For the most part, the document appears on the screen as it will be printed.
- * The width of the screen window can be adjusted between thirty and forty characters. This permits the use of a television or a monitor that overscans the screen.
- * The appearance of a document is controlled mainly by three horizontal margin settings: left margin, indent margin, and right margin. The left and indent margins can be as small as zero while the right margin setting can be as large as 213.
- * If the left margin and indent margin settings are not the same, lines that follow return characters are automatically indented (or outdented). This makes typing paragraphs and bulleted text effortless. If desired, individual lines that would normally be indented can be marked to begin at the left margin.
- * When horizontal margins are set so that complete lines will not fit on the screen, the text is automatically scrolled across the screen as you type.
- * Multiple page margins are permitted within one document. The screen can be set up to automatically adjust as you move from one page format to the next.
- * A special proofreading mode temporarily disables indenting and moves the right margin in far enough so that the text no longer has to be scrolled across the screen.
- * Dynamic word wrap instantly adjusts the display to keep lines parsed at spaces. Word wrap can be toggled on or off at any time.
- * Automatic indenting and word wrap both cause extra spaces to appear on the screen display that were never typed. These "false" spaces can be displayed as colored blocks to distinguish them from true spaces.
- * Error messages can be accompanied by a bell and/or a momentary brightening of the screen's border.

Text Entry and Editing

- * A highly visible cursor leaves no doubt where the next character typed will appear.
- * A guide bar can be turned on that extends vertically through the cursor to aid in aligning text.
- * Keys automatically repeat much more quickly than is normal for the Atari computer. This substantially reduces the time it takes to enter and edit text.
- * Two text entry modes are supported, overwrite mode and insert mode. As new text is entered in overwrite mode, it replaces the existing text. As new text is entered in insert mode, it is inserted into the existing text. The modes are distinguished by the cursor's shape.
- * The back space key always undoes what the space bar does. This is true in both overwrite mode and in insert mode.
- * Normal cursor movement using the arrow keys is supported. Additionally, the cursor can be moved up or down ten lines at a time, to the beginning or end of the current line or document, or to the top of the current screen display. The cursor can also be advanced to the next block marker, space, period, question mark, exclamation point, return character, or to the next of many special print commands.
- * As you scroll up or down the document in memory, the cursor is automatically locked at its initial horizontal position. This makes viewing the right edge of the document as easy as viewing the left edge.
- * Parts of a document can be moved or duplicated by first making deletions into a delete buffer. The contents of the delete buffer can then be inserted anywhere. Text can be appended to the delete buffer in the following ways: by character, block, all or part of a line, or from the cursor to the next space or return character.
- * Any size block of text can be marked and moved to another part of the document. Marked blocks can also be saved in separate disk files, erased, or unmarked.
- * The text in memory can be erased completely, from the beginning of the document to the cursor, or from the cursor to the end of the document.
- * The search and replace option allows for individually controlled but repeatable searches with or without replacements. Global searches with replacements can also be made. Wild card searches and replacements are supported.

- * Previously entered search and replacement specifications are provided as defaults which can be edited when new specifications are made.
- * When a search operation fails, the cursor is automatically returned to the beginning of the document. This makes it possible to continually cycle through a document with the search operation.
- * Any character (including the return character) can be searched for.
- * Any character can be entered into the text of a document. All characters can be typed in the usual way. Additionally, a short cut method of entering inverse video control characters is provided.
- * Previously entered text can be converted to upper or lower case or to inverse or regular video text.

Print Control

- * The layout of the printed page is defined with ten page format parameters.
- * Page format parameters can be set external to the document or they can be set within the document with page format command lines. Page formats can be changed repeatedly throughout the document.
- * Default formats provide one inch margins for the main text, half inch margins for the header and footer, and single spacing between lines and paragraphs. These defaults can be set by the user and permit most documents to be printed without using imbedded format commands.
- * While a document is automatically paginated according to the margin settings, individual pages can be forced to end sooner.
- * Word wrap can be left on or turned off during printing.
- * All or part of any line of text can be centered or blocked to the right margin.
- * Inverse video text can be defined within a document to represent any special type of print supported by your printer. This approach avoids using imbedded control characters which destroy horizontal spacing on the screen.
- * Four dynamically definable control characters are also supported for those who prefer them and/or when more than one group of special typefaces is needed on the same line of text.

- * A document can be printed beginning with any page. Any number of pages may be skipped between the pages that are actually printed. This allows the even and odd numbered pages of a document to be printed separately.
- * The printer can be initialized automatically at the beginning of each page. This insures that every page of the document will be printed as desired even when the document is not printed from the beginning, or when the printer is turned off between pages.
- * The pages of a document can be printed with or without waiting at the end of each page. The printing of a document can also be temporarily halted at any time from the keyboard or by an imbedded print control command character.
- * If a document consists of more than one file, the files can be chained together for printing purposes. Files can also be designated to be included within other files.
- * Graphics files and other printer code files can be designated to be printed within text files.

Disk Operations

- * The save option allows all or part of the text in memory to be stored in a disk file. The parts that can be saved are: text above the cursor, below the cursor, or within a marked block.
- * The load option allows a previously saved text file to be recalled. The file being loaded can be inserted into or appended onto the document in memory, or it can overwrite all or part of the document in memory.
- * Both the save and load operations provide the last used file specification as a default.
- * A new version of the word processing program can be saved. This version will use all the existing settings as defaults.
- * This word processor can be used to edit any type of file with any legal file name. No special headers are expected during load operations, no extra information is added when a file is saved, and all 256 ASCII characters are processed properly.
- * A special feature called view/verify displays any size disk file on the screen without altering the text in memory.
- * The usual DOS functions (directory, erase, format, rename, write protect and unprotect) are supported.

GETTING STARTED

Equipment You Will Need

The Rewrite Word Processor is designed for use with any of the 8-bit Atari home computers with at least 32K of memory. To use this program you will also need:

- * A television or monitor.
- * A disk drive with extra diskettes.
- * A printer for printing documents.
- * A cassette program recorder (optional)

Making a Back Up Copy of Your Program Diskette

It is good practice to use a copy of the program diskette you received. The original should be kept in a safe place. This word processor is compatible with Atari DOS 2.0S. If you wish to use this version of DOS you should use the DUPLICATE DISK option from the DOS menu to make an exact copy of your original diskette.

If you prefer to use this word processor with a different version of DOS you will need to follow the directions in your DOS manual for converting DOS 2.0S files. The word processing program is contained in the AUTORUN.SYS file.

A procedure for saving a different version of this word processor is described later in this manual. (See **Save File** in **THE OPTION MENU**, pages 14-15.)

Loading the Program

To load the program:

- * Make sure the computer is off.
- * Remove all ROM cartridges from the computer.
- * Turn the disk drive on.
- * Turn the television or monitor on.
- * Put the program diskette in the disk drive.
- * Turn the computer on.
- * Wait for the program to load.

ESSENTIAL INFORMATION

The Screen Display

When the Rewrite Word Processor starts, a 40-column by 20-line display will appear on the screen with this message at the top:

The Rewrite Word Processor Ver 2.4
By: Robert Ertl Copyright 1987

The top two lines are special lines and are separated from each other and from the rest of the screen. The top line is called the prompt line. This is where all of the input prompts will appear. The line below the prompt line is called the input line. All responses to the prompts are entered on the input line. Occasionally, a double line secondary menu will occupy both the prompt line and input line.

When the first key is pressed, the title message is replaced with:

```

MENUS: [START] [SELECT] [OPTION]
..!0...!.....!10...!.....!20...!.....!30...!....

```

The top line is a reminder that the three function keys (START, SELECT, and OPTION) provide entry into the three main menus. This prompt is called the default prompt because it appears on the prompt line most of the time. The input line displays a scrolling scale that shows where the text will appear on the printed page. The default tab stops are marked just beneath the scrolling scale. Below the tab stop markers is an eighteen line window into the text memory area. The text is generally displayed in the text window the way it will appear on paper when it is printed.

Symbols for Keys and Key Combinations

Capital letters enclosed in square brackets are used to indicate keys that are to be pressed. For example:

```

[RETURN] means press the RETURN key.
[OPTION] means press the OPTION key.

```

In the three main menus, the key that is to be pressed to choose an item from that menu appears in front of the description in square brackets. For example:

[D] DISK DIRECTORY

This means that the D key should be pressed to obtain a disk directory.

In the secondary menus the first letter of the description is enclosed in square brackets. The search and replace menu is shown below as an example:

[G]lobal S&R [S]earch [R]eplace

To choose search in this menu, the S key should be pressed.

When a group of keys are joined together with the plus symbol between them, the first key (or first two keys) should be held down while pressing the last key. Some examples follow:

[SHIFT]+[TAB]means hold the SHIFT key down while pressing the TAB key.
 [CTRL]+[INSERT]means hold the CONTROL key down while pressing the INSERT key.
 [SELECT]+[P]means hold the SELECT key down while pressing the P key.
 [BREAK]+[M]means hold the BREAK key down while pressing the M key.
 [SHIFT]+[CTRL]+[DELETE] ..means hold both the SHIFT key and the CONTROL key down while pressing the DELETE key.

When a group of keys appear together without the plus symbol between them, each key should be pressed and then released before the next key is pressed. Some examples follow:

[SELECT] [P]means the SELECT key should be pressed first and released. After this, the P key should be pressed and released.
 [BREAK] [M]means the BREAK key should be pressed first and released. After this, the M key should be pressed and released.
 [OPTION] [L] [T]means that the OPTION key should be pressed and released. After this, the L key should be pressed and released. Finally, the T key should be pressed and released.

The Three Main Menus

When one of the function keys is pressed, the menu of the same name is displayed in the text window area and the scrolling scale is replaced by the title of the menu. The default prompt remains on the prompt line. The OPTION menu has to do with controlling peripheral devices. The SELECT menu allows the user to select settings or specifications. The START menu is used to carry out editing operations. These three menus are shown on the next page:

MENUS: [START] [SELECT] [OPTION]
OPTION PERIPHERAL OPTIONS MENU

[D] DISK DIRECTORY
[E] ERASE FILE
[F] FORMAT DISK
[L] LOAD FILE
[P] PRINT DOCUMENT
[R] RENAME FILE
[S] SAVE FILE
[U] UNPROTECT FILE
[V] VIEW/VERIFY FILE
[W] WRITE-PROTECT FILE

MENUS: [START] [SELECT] [OPTION]
SELECT SETTINGS OR CHECK SIZE MENU

[B] BLOCK BEGINNING & END
[D] DISPLAY COLOR & WIDTH
[F] FREE MEMORY
[K] KEYSTROKES BEFORE SAVE
[L] LINEFEED WITH RETURN
[P] PAGE FORMAT PARAMETERS
[R] REPLACEMENT SPECIFICATION
[S] SEARCH SPECIFICATION
[T] TOGGLE MODES

MENUS: [START] [SELECT] [OPTION]
START EDITING OPERATIONS MENU

[B] BLOCK OPERATIONS
[D] DELETE TEXT TO BUFFER
[E] ERASE TEXT
[I] INSERT OPERATIONS
[S] SEARCH AND/OR REPLACE
[T] TEXT CONVERSION

When a menu item is chosen, the menu disappears and the window into the text memory is restored. The prompt or secondary menu for the item chosen will then appear at the top of the screen.

Responding to Prompts

A prompt appears on the prompt line when something is to be entered on the input line. Quite often a default input is provided on the input line. The default may be used by simply pressing the RETURN key. If desired, defaults can be prevented from appearing on the input line by keeping the function key pressed while choosing the menu item that results in the prompt. All entries on the input line, including a default entry, can be edited using the input editing keys which are described later in this manual. (See **Special Input Line Key Commands** in **KEY COMMANDS**, pages 35-36.)

Device Names and File Specifications

Some prompts require device names or file specifications to be entered. In these prompts, the information enclosed in bent brackets describes what is to be typed.

When <Dev> appears it means that a device name needs to be entered. The following is a list of the default device names.

K:	Keyboard
S:	Screen
E:	Editor (Screen and Keyboard)
P:	Printer
C:	Cassette Program Recorder

All device names end with a colon. When a device name is requested in a prompt, the colon does not have to be typed.

When <Filespec> appears in a prompt, it means a disk file specification needs to be entered. A disk file specification begins with a disk device name. The following are examples of disk device names that are recognized by most of the disk operating systems that are available for the Atari computer:

D:	Disk drive number one
D1:	Disk drive number one
D2:	Disk drive number two
D3:	Disk drive number three
D4:	Disk drive number four

If the disk device name is omitted from the file specification in response to a <Filespec> prompt, the device name "D1:" will be assumed. Disk device names must be included when entering a file specification in response to a <Dev> prompt.

The device name in a file specification must be followed by a filename, and optionally, a filename extender. A filename consists of up to eight alphabetic or numeric characters. The first character must be an alphabetic character. Extenders consist of up to three alphabetic or numeric characters. If an extender is used, it must be separated from the filename with a period. Some examples of acceptable disk file specifications are given on the next page:

```
D:FILENAME.EXT
D1:TEXTFILE.5
D2:MYBOOK.CH4
D:BASIC.LST
NOTES
D2:AUTORUN.SYS
```

Some disk operations do not need a filename. In this case, <#> will appear in the prompt as one of the options. This means that only the number of the desired disk drive needs to be entered.

Using the BREAK Key

Any input may be aborted at anytime by pressing the BREAK key. The default provided for an aborted input remains as it was before the input began. Any menu may also be exited by pressing the BREAK key. The BREAK key will abort any operation that is in progress. With many operations the following prompt appears periodically to allow an operation to be continued or terminated:

Push any key... [BREAK] to exit.

Pressing the BREAK key in response to this prompt will halt the operation, while pressing any other key will allow the operation to continue.

The BREAK key has a special function in the main text window. When the BREAK key is pressed, it causes the next character typed to be converted to an inverse video control character. This character is then inserted into the text at the cursor position. Such characters appear as underlined upper case letters on the screen. These characters are used as special command characters by this word processor. For example, the special character used to mark the beginning and end of a block of text (inverse video control M) could be typed in the usual way:

```
[INVERSE-VIDEO-KEY] [CTRL]+[M] [INVERSE-VIDEO-KEY]
```

It can also be typed, however, with:

```
[BREAK] [M] or [BREAK]+[M]
```

The BREAK key may be pressed prior to typing the M key or while typing the M key. This abbreviated method only works when entering text into the main text window. As mentioned above, pressing the BREAK key while responding to a prompt will abort the input that was in progress.

THE OPTION MENU

If the OPTION key is pressed, the following menu is displayed:

**MENUS: [START] [SELECT] [OPTION]
OPTION PERIPHERAL OPTIONS MENU**

**[D] DISK DIRECTORY
[E] ERASE FILE
[F] FORMAT DISK
[L] LOAD FILE
[P] PRINT DOCUMENT
[R] RENAME FILE
[S] SAVE FILE
[U] UNPROTECT FILE
[V] VIEW/VERIFY FILE
[W] WRITE PROTECT FILE**

These menu items direct the use of peripheral devices such as printers or disk drives. Any time an error condition is generated by a peripheral device the following error message is displayed along with an error code number.

Error code:

Some of the most common error codes that may appear while using this word processor with Atari DOS 2.0S, are listed below with non-technical explanations. See your DOS manual for the complete list of error codes.

130 A non-existent device was specified.
138 The device does not respond.
139 The device cannot respond.
144 The diskette cannot be written on because:
* The disk drive door is open.
* The diskette is write protected.
* The diskette has a bad sector.
146 The device cannot do what was requested.
160 A non-existent disk drive was specified.
162 The diskette is full.
164 The file cannot be read.
165 An illegal filename was specified.
166 The file is locked.
169 The disk directory is full (64 filenames).
170 The specified file cannot be found.
173 The diskette cannot be formatted.

Disk Directory

The first item in the OPTION menu is disk directory. When [D] is pressed, the following prompt appears on the prompt line:

Directory <#> <Filespec> or [RETURN]

The parts of the prompt that are enclosed in bent brackets (<>) contain descriptions of what is to be typed on the input line. If a number is entered, the complete directory of that numbered drive is displayed. If a file specification is entered, all files that match the file specification are displayed. File specifications that do not include a device name will default to disk drive number one. Standard Atari file naming conventions are used and wild card characters are permitted, (See your DOS manual for details about wild card usage.) If the RETURN key is pressed without typing anything else, the disk directory from disk drive number one is displayed.

Format Disk

Choice [F] in the OPTION menu has a prompt that is very similar to the directory prompt.

Format <#> or <Filespec>

When a diskette is formatted, all of the information on it is erased. Pressing just the RETURN key here does not carry out this operation on disk drive number one, as it did in the directory operation above. Pressing the RETURN key, without first entering a number or a file specification, causes the format operation to be aborted. This helps prevent a diskette from accidentally being formatted.

View/Verify File

Pressing [V] while in the OPTION menu causes the following prompt to appear on the prompt line:

View/Verify <Filespec>

This operation displays the specified file one screen at a time without losing the file currently in memory. With this prompt, the file specification used for the last save or load operation appears on the input line as a default. (See **Save File** and **Load File** in **THE OPTION MENU**, pages 14-15.) This default can be used by simply pressing the RETURN key. If a default is not desired, it can be eliminated with [SHIFT]+[DELETE] or kept from even appearing by keeping the OPTION key down while pressing [V]. The default can also be edited using the input editing keys. (See **Special Input Line Command Keys** in **KEY COMMANDS**, pages 35-36.)

Erase, Write Protect & Unprotect

The following disk operations all have similar prompts:

Erase <Filespec>
Write Protect <Filespec>
Unprotect <Filespec>

These operations do not provide a default on the input line. The erase operation removes the specified file(s) from a diskette. If a file is write protected it is not possible to alter it or erase it. The unprotect operation removes write protection. (See your DOS manual for further details.)

Rename File

Renaming a file requires both the file's original name and its new name to be entered. The prompt follows:

Enter <Old Filespec>,<New filename>

If the file that is being renamed is not on disk drive number one, the appropriate disk device name needs to be specified as part of the old file specification. The file's new name should not include a device name. For example, the following will rename the file on disk drive number two called OLDSTORY as NEWBOOK:

D2:OLDSTORY,NEWBOOK

The following is all that is needed to make the same change on disk drive number one:

OLDSTORY,NEWBOOK

Save File

The save operation in the OPTION menu has many options, as shown in its double line menu below:

SAVE [A]ll [B]lock [W]ord processor
From Cursor [D]own [U]p

Depending upon which choice is made, one of the following prompts will appear:

Save <Filespec>
Save Block <Filespec>
Save Word processor <Filespec>
Save Cursor...End <Filespec>
Save Top...Cursor <Filespec>

All of the choices, except for [W], have to do with saving all or part of the text currently in the text memory area. If a file has previously been loaded or saved, the file specification used at that time would be provided here as a default on the input line. This specification may be used by simply pressing the RETURN key. It may also be edited, cleared from the input line, or avoided as explained earlier.

Choice [W] saves a new version of the word processing program. This new version will use the current settings for each of the following as defaults:

- * The last save or load text file specification used
- * Display color
- * Display luminance
- * Display width
- * Keystrokes before SAVE reminder
- * Linefeed ASCII value needed
- * The ten page format parameters
- * The last replacement specification defined
- * The last search specification defined
- * Automatic page formatting toggle
- * False spaces toggle
- * Guide bar toggle
- * Insert mode toggle
- * Type of error bell
- * Word wrap toggle

The default save or load text file specification is set by loading or saving a file. Initially this default is null. A null default can be set again by attempting to load or save a text file using a null file specification. The rest of the defaults can be set using the SELECT menu. (See **THE SELECT MENU**, pages 17-22.)

After all the desired default settings have been made, the word processing program should be saved on a diskette that contains the DOS files. (See your DOS manual on how to save the DOS files on a diskette). When using DOS 2.0S, the file specification should be AUTORUN.SYS.

Load File

The load operation in the OPTION menu has multiple options and produces this double line menu:

**LOAD [I]nsert at cursor [A]ppend
Overwrite from: [T]op or [C]ursor**

Depending upon which choice is made, one of the following prompts will appear:

**Insert Load <Filespec>
Append Load <Filespec>
Overwrite Load <Filespec>**

As with the save operation described above, the file specification used for the most recent save or load appears on the input line as a default. The insert load operation inserts the specified file at the current cursor position without losing any of the existing document in memory. The append operation loads the specified file beginning at the end of the existing document in memory, regardless of where the cursor is positioned. The overwrite from top operation causes the text memory to be cleared before the specified file is loaded. The overwrite from cursor operation loads the specified file beginning at the cursor position. In this case, the text that followed the cursor before the new file was loaded is lost.

Print Document

The print option requires a device name and allows for two optional parameters. The prompt follows:

PRINT <Dev>,<Init Skip>,<Reg Skip>

The normal device used for printing a document is a printer. The device name for most printers is P:. Documents may also be printed on the screen. This can be done by using S: or E: for the device name. It is also possible to print a document to a disk file. To do this, a disk file specification should be used. Since a disk file specification was not specifically requested by this prompt, such a specification must include the proper disk device name. (See **Device Names and File Specifications** in **ESSENTIAL INFORMATION**, page 10.)

The optional parameters, if used, must be separated from the specified device and from each other with commas. The first optional parameter is the initial number of pages of the document that are to be skipped before a page is actually printed. The second optional parameter is the regular number of pages that are to be skipped between printed pages once the first page is printed. If no values are entered for these parameters, they default to zero. This means that no pages are to be skipped, so every page will be printed.

The general rule to follow when entering values for these parameters is: The value that follows the first comma will be used for the first parameter and the value that follows the second comma will be used for the second parameter. This rule allows either one, both, or neither of the parameters to be specified. For example:

Input	Pages that will be printed
P:	1,2,3,4,...
P:,1	2,3,4,5,...
P:,1,1	2,4,6,8,...
P:,0,1	1,3,5,7,...
P:,,1	1,3,5,7,...
P:,4	5,6,7,8,...
P:,4,4	5,10,15,...

As each new page is printed, the following message appears along with the current page number.

Printing Page:

The printing process may be temporarily interrupted with [CTRL]+[1]. Pressing this key combination again resumes the printing process.

THE SELECT MENU

The SELECT function key produces the menu:

```

MENUS:  [START] [SELECT] [OPTION]
SELECT  SETTINGS OR CHECK SIZE MENU

[B]      BLOCK BEGINNING & END
[D]      DISPLAY COLOR & WIDTH
[F]      FREE MEMORY
[K]      KEYSTROKES BEFORE SAVE
[L]      LINEFEED WITH RETURN
[P]      PAGE FORMAT PARAMETERS
[R]      REPLACEMENT SPECIFICATION
[S]      SEARCH SPECIFICATION
[T]      TOGGLE MODES

```

All of these choices have to do with making selections except for the free memory option.

Free Memory

The free memory choice displays the number of additional characters that will fit in the text memory. The following is displayed on a 48K system before any text is entered:

Free Memory: 22272

Block Beginning & End

Choice [B] allows a block of text to be marked or unmarked. If the block is to be marked, the cursor must be positioned at either the beginning or the end of the block before using this option. The one-line menu for this item follows:

[M]ark Beginning/End [U]nmark Block

When the [M] is pressed, a special character (inverse video control M) is inserted into the text at the cursor position. This character appears as an underlined capital M on the screen. This same special character is used for marking the beginning as well as the end of a block. If more than two such characters appear in a document, the first two are used during block operations. (See **Block Operations** in **THE START MENU**, page 23.)

When the [U] is pressed, all of the block markers are removed from the document, beginning at the top. Just before each one is removed, it is displayed on the screen. If desired, the BREAK key may be used to abort this process before all of the markers are removed.

There are alternate methods of accomplishing the same tasks for many of the items in both the SELECT and START menus. As these are discussed, the alternative methods will be noted. For example, a block can be marked simply by typing the inverse video control M character at the beginning and at the end of the text that is to be marked as a block. This is normally done on the Atari computer as follows:

[INVERSE-VIDEO-KEY] [CTRL]+[M] [INVERSE-VIDEO-KEY].

Since this special character, as well as each of the print control command characters discussed later, is an inverse video control character, a quick method of inserting it into the text has been devised. The inverse video control M character may be entered as follows:

[BREAK] [M] or [BREAK]+[M]

The BREAK key may be pressed prior to pressing the [M] key or it may be held down while the [M] key is pressed. This use of the BREAK key does not work when entering information on the input line. In that case, pressing the BREAK key will abort whatever input is in progress.

Display Color & Width

Choice [D] from the SELECT menu allows the screen display to be adjusted in both size and color. The double line menu is:

DISPLAY [**<**]Color[**>**] [**N**]arrow
SETTINGS [**L**]uminance [**W**]iden

There are sixteen colors to choose from. The luminance choice toggles between light and dark backgrounds. The narrow and widen options make it possible to adjust the width of the word processor's display. This allows this word processor to be used with a television that overscans the screen.

Keystrokes Before Save

The number of keystrokes that will be permitted before a SAVE reminder is displayed can be set by choosing item [K] in the SELECT menu. The default value is 5000. Any value between 1 and 9999 may be entered. The prompt looks like this:

Keystrokes before SAVE reminder...

When the specified number of keystrokes has been typed, the error bell rings and the following prompt becomes the default prompt:

It's time to SAVE your file!

This new prompt remains the default prompt until the file is saved or the number of keystrokes is reset.

Linefeed with Return

Item [L] in the SELECT menu allows the user to specify the ASCII value of the linefeed character that the printer expects (if any) with each carriage return. If when printing a document the paper does not advance, this value needs to be set. For most printers it should be set to ten, which is the ASCII value for linefeed. The default value of zero indicates that no linefeed character is needed. The prompt is:

Linefeed ASCII needed with RETURN

Page Format Parameters

Specification of the general page format is selected with item [P] in the SELECT menu. This produces the following prompt.

ENTER: Lf,In,Rg,Ls,Ps,Hd,Tp,Bt,Ft,Ln

From left to right, these page format parameters are: left margin, indent (or outdent) margin, right margin, line spacing, paragraph spacing, header position, top margin, bottom margin, footer position, and length of page. When this prompt appears, the current parameter settings appear on the input line. Not all of the parameters have to be entered each time. Parameters at the end of the list may be omitted. Any parameter at the beginning of the list may be skipped by entering the comma that would normally follow it.

The first three parameters--left margin, indent margin, and right margin--are actually horizontal print positions. These are measured in characters from the left-most print position on the paper, which is considered print position zero. The left margin and indent margin positions may be assigned any values from 0 to 195. The right margin position must be greater than both of these, but no larger than 213. Whenever horizontal page formats are chosen that are not compatible, the following error message will appear:

Command Syntax or Parameter Error!

Most lines begin at the left margin. All lines which follow a return character automatically begin at the indent margin. Such lines can be forced to begin at the left margin by prefixing them with the special character (inverse video control L) produced by [BREAK] [L]. (See **Horizontal Alignment Commands** in **IMBEDDED PRINT CONTROL COMMANDS**, page 37.) These lines are accurately displayed on the screen.

Paragraphs are automatically indented when the left margin is set to a smaller value than the indent margin. Numbered or bulleted blocks of text are formatted automatically when the indent margin is set to a smaller value than the left margin. In this case, the indent margin is actually used as an outdent margin.

Lines are normally ended at spaces before the right margin setting is reached. If this causes a line to begin with the second of two consecutive spaces, both spaces are put at the end of the previous line.

The next two page format parameters, line spacing and paragraph spacing, determine the number of lines the paper is to be advanced: the paragraph spacing value is used after lines that end with a return character and the line spacing value is used the rest the time.

The next four page format parameters--header position, top margin, bottom margin, and footer position--are vertical page positions, measured in lines past the first line on the page. These may be assigned values between 0 and 255. Each of the last three of these vertical parameters should be assigned a value that is no less than the value assigned to the previous.

The header position parameter is the number of the line that the header is to be printed on. The top margin parameter is the number of the line that the first line of the main text is to be printed on. Each page is automatically ended before the bottom margin position is reached. The footer position parameter is the number of the line that the footer is to be printed on.

The length of page is the final page format parameter. It should be assigned a value that is equal to the total number of lines available on the paper used. A standard linefeed is one-sixth of an inch, so for eleven inch paper, this value should be set to 66.

The original default settings are: 5,10,75,1,1,3,6,60,63 & 66. With pica print and 1/6th inch line spacing these provide three-quarter 3/4 inch horizontal margins for the main text with paragraphs indented five spaces, one-half inch vertical margins for the header and footer, and single spacing between both lines and paragraphs.

Page formats that are on the input line may be imbedded within the text of a document at the cursor position with [SHIFT]+[INSERT]. Page formats that are imbedded within the text can be activated by using a special input line feature that allows characters from the main text to be copied up to the input line. This is done with [CTRL]+[UP-ARROW]. (See **Special Input Line Key Commands** in **KEY COMMANDS**, pages 35-36 for more details.) Page formats in the text can also be designated to be automatically activated whenever the cursor is positioned on them. This automatic feature can be turned on through the toggle modes item in the SELECT menu (see page 22).

A special proofreading screen can be set up by entering a blank line in response to the SELECT menu's page format prompt. A blank input line can be easily obtained with [SHIFT]+[DELETE]. A blank line can also have be obtained by keeping the [SELECT] key down while pressing [P] to choose the page format item. This keeps the current page formats from being displayed on the input line as defaults.

The proofreading screen uses zero for the left margin and indent margin settings, and a right margin setting that is small enough so that the lines will fit on the screen display. The normal screen can be restored by pressing [SELECT], [P] and [RETURN] separately. This activates the page formats that were in use before the proofreading screen was set up.

Search & Replacement Specifications

Search and replacement specifications are defined using [S] & [R] in the SELECT menu. Previous specifications appear as defaults unless the SELECT key is held down to suppress them. The question mark is treated as a wild card character when used in either specification. (See **Search and/or Replace** in **THE START MENU** for more details on wild card characters, page 26.) The prompts for defining the search and replacement specifications are:

Search for...

Replace with...

The feature that allows text to be copied up from the main text area to the input line is useful when defining search or replacement specifications which already appear within the main text. This is done with [CTRL]+[UP-ARROW]. As with all inputs, any of the 256 ASCII characters can be entered on the input line. Special editing characters like back space and the arrow keys can be entered by pressing the escape key prior to typing the desired key. This also works for the return key!

Toggle Modes

The last item on the SELECT menu allows several modes to be toggled. The double line menu is:

[A]UTO-P [F]alse spaces [G]uide bar
[I]nser^t [E]rror bell [W]ord wrap

Word wrap is the process of parsing the continuous text in the text memory into lines without splitting words. Turning word wrap off causes lines to be parsed at the right margin regardless of whether or not this causes a word to be split. This is sometimes useful when working on a proofreading screen or when working with data files. It determines whether or not word wrap is activated for the screen display as well as when printing a document. The alternative method of toggling word wrap is [SHIFT]+[CTRL]+[W]

Automatic indenting and word wrap both cause spaces to appear on the screen display that were not typed and which are really not part of the document. Pressing [F] toggles the appearance of these "false" spaces between colored blocks and regular spaces. The alternative method of toggling false spaces is [SHIFT]+[CTRL]+[F].

Pressing [G] in this menu toggles a guide bar which consists of a pair of vertical lines that extend the height of the screen through the left and right edges of the cursor. This eliminates any guesswork about vertical alignment. The alternative method of toggling the guide bar is [SHIFT]+[CTRL]+[G].

Choice [I] in this menu toggles insert mode. When insert mode is off the cursor appears as a flashing character superimposed over a slightly larger rectangle. As each new character is typed, it replaces the character at the cursor position. This is the default mode for this word processor and is often referred to as overwrite mode. When insert mode is on, new text is inserted into the existing text as it is typed. In this mode, the cursor appears as a flashing character with a vertical bar in front of it. The alternative method of toggling insert mode is [SHIFT]+[CTRL]+[I]

There are two situations when, even though text is being entered in overwrite mode, the characters typed will actually be inserted. The first is when the BREAK method of entering inverse video control characters is used. The second is when the cursor is on a return character.

The error bell is momentarily activated whenever an error message is displayed. It may be audio, visual, audio and visual, or it may be turned off. Each time [E] is pressed from this menu, the next of these four options is demonstrated and set for use. The bell sound that is normally made by the Atari computer when [CTRL]+[2] is pressed is used for the audio error bell. When the visual error bell is activated the border around the screen display abruptly changes from dark to light and then fades back to dark. The alternative method of choosing the desired error bell option is [CTRL]+[2].

Choice [A] in this menu toggles the automatic page formatting feature. When turned on, page format parameters within the text are automatically activated whenever the cursor is positioned on a page format command character (inverse video control P). When this happens, the page format command line is positioned at the top of the screen. Pushing [A] rings the bell when this feature is being turned on. The absence of a bell indicates that the feature is being turned off. There is no alternative method of toggling this mode.

THE START MENU

The START menu contains items that carry out editing operations.

```

MENUS:  [START] [SELECT] [OPTION]
START   EDITING OPERATIONS MENU

[B]      BLOCK OPERATIONS
[D]      DELETE TEXT TO BUFFER
[E]      ERASE TEXT
[I]      INSERT OPERATIONS
[S]      SEARCH AND/OR REPLACE
[T]      TEXT CONVERSION

```

Block Operations

We have already seen one block operation in the OPTION menu with which the text in a marked block can be saved. Choice [B] in the START menu includes this operation along with other block operations, as its double line menu indicates below:

```

BLOCK    [M]ove to Cursor  [S]ave
           [U]nmark [D]elete to Buffer [E]rase

```

Choice [M] will move a marked block to the current cursor position. (A block can be marked using the SELECT menu.) This feature is especially powerful in that any size block can be moved. There are two limitations to the block move operation. First, the delete buffer (described below) is used and cleared. Secondly, the desired destination of the block is marked with the special character that has an ASCII value of 255. If such characters already exist in the document, the first one will be replaced with the marked block.

Choice [D] deletes the text within the marked block into the retrievable delete buffer, as described below. Choice [E] will erase any size block that is marked. Once text is erased in this manner it cannot be retrieved. Unmark removes the first two block markers from the text memory. It is the same as [SELECT] [B] [U] mentioned above. Anytime a block operation is attempted without first marking a block the error message will appear:

Block must first be marked!

Delete Text to Buffer

Choice [D] in the START menu allows for various kinds of deletions. All deleted text is appended to the delete buffer from which it can later be retrieved. The prompt is:

```

DELETE [B]lock [( )--[( )] [C]hr [L]ine
          until [S]pace [R]eturn [Z]ap buffer

```

Choice [B] deletes the first marked block in the text memory. It is the same as [START] [B] [D] mentioned above. Unlike the other choices in the delete menu, this operation first clears the delete buffer. All of the other delete operations leave the contents of the delete buffer intact. Choice [Z], zap buffer, can be used at any time to clear the delete buffer.

Choice [()] deletes from the beginning of the line that the cursor is on to the current cursor position. The alternate method of performing this deletion is [CTRL]+[()].

Choice [)] deletes from the current cursor position to the end of the line that the cursor is on. The alternate method of performing this deletion is [CTRL]+[)].

Choice [C] deletes the character to the right of the cursor. [CTRL]+[DELETE] performs the same operation, except that it does not append the deleted character to the buffer.

Choice [L] deletes the line that the cursor is on. The alternate method of performing this deletion is [SHIFT]+[DELETE]

Choices [S] and [R] delete text from the cursor position to the next space or return character, respectively. The alternate methods of performing these last two deletions are [CTRL]+[SPACE] and [CTRL]+[RETURN], respectively.

Anytime an attempt is made to append more text to the delete buffer than there is room for, this message will appear:

Not enough room in buffer!

When the alternate methods of deleting text are used the buffer is first cleared, unless the previous keystroke was also a deletion. This facilitates automatic clearing of the buffer while permitting consecutive deletions to transfer large blocks of text to the buffer. The contents of the buffer may then be inserted anywhere else in the document using [START] [I] [B] or [SHIFT]+[INSERT] as explained below. (See **Cut and Paste Operations** in **KEY COMMANDS**, page 32.)

Erase Text

Choice [E] in the START menu allows for erasing text. The term erase, unlike delete, means that once text is removed it cannot be retrieved. The double line menu is:

**ERASE TEXT: [A]ll [B]lock
From Cursor [D]own [U]p [S]paces**

Choice [A] erases all of the text in the text memory. Choice [B] erases the text within a marked block. Choices [D] and [U] erase all the text below and above the cursor, respectively. Choice [S] will remove a contiguous block of spaces beginning at the cursor position. The alternate method of doing this is [SHIFT]+[CTRL]+[DELETE].

Insert Operations

Choice [I] in the START menu produces this double line menu:

```
INSERT [B]uffer [L]oad file
[S]pace [A]ll available space
```

Choice [B] inserts the contents of the delete buffer at the cursor. The alternate method of performing this insertion is [SHIFT]+[INSERT]. Anytime an attempt is made to insert the contents of an empty delete buffer into the text memory, this message will appear:

Buffer is empty!

When an attempt is made to insert more text into the text memory than there is room for, the following message will appear:

Text memory is full!

Choice [L] performs an insert load operation which inserts the contents of the specified file at the cursor position. This is the same as [OPTION] [L] [I] described above.

Choice [S] inserts a screenful of spaces. If there is less than a screenful of text memory available, this will insert as many spaces as will fit. The alternate method of performing this insertion is [SHIFT]+[CTRL]+[INSERT].

Choice [A] inserts all available space in the text memory at the cursor position. There is no alternate method of performing this operation.

Search and/or Replace

The search and replace option in the START menu allows for single searches, single replacements and global searches with replacements, as shown by this one line menu:

[G]lobal S&R [S]earch [R]eplace

Pressing the [S] key will cause the document to be searched for a match with the search specification. (See **Search & Replacement Specifications** in **THE SELECT MENU**, page 21.) All searches begin with the first character after the cursor. If a search fails, the cursor is positioned at the top of the document.

One of three things will happen when the [R] key is pressed. If the cursor is positioned at a portion of the document that matches the search specification, that matching text will be replaced with the replacement specification text. If the search specification is null, the replacement specification text will be inserted into the document at the cursor position. When neither of these conditions exist, a single search will be executed.

The question mark character may be used as a wild card character in the search and/or replacement specifications. During a search, any character in the document will be considered a match for corresponding question marks in the search specification. Question marks are treated as wild card characters in the replacement specification only when the search and replacement specifications are the same size. In this case, characters in the document which correspond to the question marks in the replacement specification are not altered. Wild card characters allow similar items in a document to be replaced with the same item or to be altered in similar ways.

Pressing [G] first allows the search and replacement specifications to be verified and possibly modified. Then it has the same effect as pushing the [R] key repeatedly until no match is found. Each time a replacement is made, it is displayed on the screen. Pressing the BREAK key during a global search with replacement will halt the operation.

The alternate method of performing a single search is [SHIFT]+[CTRL]+[S]. The alternate method of performing a single replacement is [SHIFT]+[CTRL]+[R]. There is no alternate method of performing a global search with replacement.

Text Conversion

The last option in the START menu allows for easy conversion of previously entered text.

**TEXT TO: [U]ppercase [I]nverse
 [L]owercase [R]egular**

Each time one of the above keys is pressed, the character at the cursor is converted appropriately and the cursor is moved to the next character.

KEY COMMANDS

Some keys or, combinations of keys, cause an operation to be carried out rather than text to be entered. These keys or key combinations are called key commands. Some of the key commands are alternate methods of performing menu operations which were described in the previous three sections. In these cases, the key sequence used in the menu is given.

Screen Display Settings

The following key commands toggle settings that affect the appearance of the screen display.

[SHIFT]+[CTRL]+[G]	This toggles the guide bar. When on, the guide bar appears as a pair of lines that extend vertically the height of the screen through the left and right edges of the cursor. This aids in aligning text into columns. The guide bar can also be toggled with [SELECT] [T] [G].
[SHIFT]+[CTRL]+[W]	This toggles word wrap. When on, word wrap prevents words from being split at the right margin. The word wrap toggle affects both the screen display as well as the printed document. Word wrap can also be toggled with [SELECT] [T] [W].
[SHIFT]+[CTRL]+[F]	This toggles the display of false spaces. Both word wrap and automatic indenting cause spaces to appear that were not typed. These false spaces can be displayed as colored blocks. The display of false spaces can also be toggled with [SELECT] [T] [F].
[CTRL]+[2]	The error bell is momentarily activated whenever an error message is displayed. It may be audio, visual, audio and visual, or it may be turned off. Each time this key command is given, the next of these four options is demonstrated and set for use. This can also be done with [SELECT] [T] [E].

Keyboard Settings

Each of the following key commands has an effect on what happens when other keys are pressed during text entry and editing.

[SHIFT]+[CTRL]+[I]	This toggles between insert mode and overwrite mode. Overwrite is the default mode. In this mode, new text replaces existing text as it is typed. The cursor appears as a flashing character superimposed on a slightly larger rectangle. In insert mode, the new text is inserted into the existing text as it is typed. The cursor appears as a flashing letter with a small vertical bar in front of it. This toggle can also be implemented with [START] [T] [I].
[CAPS]	This toggles the keyboard between upper case and lower case entry for letters which determines whether letters typed without [SHIFT] or [CTRL] will be entered in upper case or lower case. If this key is pressed after the keyboard has been set for control characters, it returns the keyboard to lower case entry. This key is labeled [CAPS/LOWR] on some keyboards.
[SHIFT]+[CAPS]	This sets the keyboard for entry of upper case letters, when keys are pressed without [SHIFT] or [CTRL]. (See [CAPS] above for more details.)
[CTRL]+[CAPS]	This sets the keyboard for entry of control characters, when keys are pressed without [SHIFT] or [CTRL]. (See [CAPS] above for more details.)
[INVERSE]	This key toggles the keyboard between regular video and inverse video entry. It is labeled with the Atari logo on some keyboards and with light and dark triangles on others.
[SHIFT]+[TAB]	This sets a tab stop at the current cursor column. The tab stop markers appear on the screen between the scrolling scale and the text window.

[CTRL]+[TAB]	This clears the tab stop at the current cursor column.
[SHIFT]+[CTRL]+[TAB]	This clears all of the tab stops.
[ESC]	Pressing this key causes the next regular keystroke to produce a character. This allows some of the keys that normally carry out an operation to be typed as characters.
[BREAK]	Pressing this key causes the next regular keystroke to produce an inverse video control character. This provides an easy method of entering the special command characters used by <u>Rewrite</u> .

Cursor Movement

The following key commands cause the cursor to be moved within the text memory.

[CTRL]+[UP]	This moves the cursor up one screen line, while maintaining the original cursor column and horizontal position of the text window. If the cursor is on the first line of the text memory it will not be moved.
[CTRL]+[DOWN]	This moves the cursor down one screen line, while maintaining the original cursor column and horizontal position of the text window. If the cursor is on the last line of the text memory it is moved to the last character that has been entered into the text memory.
[CTRL]+[LEFT]	This moves the cursor one character to the left. If the cursor is at the beginning of a line it will be moved to the end of the previous line. If the cursor is at the beginning of the text memory, it will not be moved.

[CTRL]+[RIGHT]	This moves the cursor one character to the right. If the cursor is at the end of a line it will be moved to the beginning of the next line. If the cursor is positioned on the last character that has been entered into the text memory, it will not be moved.
[SHIFT]+[CTRL]+[UP]	This has the same effect as pressing [CTRL]+[UP] ten times.
[SHIFT]+[CTRL]+[DOWN]	This has the same effect as pressing [CTRL]+[DOWN] ten times.
[SHIFT]+[CTRL]+[([	This moves the cursor to the beginning of the current line.
[SHIFT]+[CTRL]+[)]	This moves the cursor to the end of the current line.
[CTRL]+[CLEAR]	This moves the cursor to the beginning of the text memory.
[SHIFT]+[CLEAR]	This moves the cursor to the last character that has been entered into the text memory.
[SHIFT]+[CTRL]+[CLEAR]	This moves the cursor to the top left corner of the text window.
[TAB]	This advances the cursor to the next tab stop. If the cursor is at the end of a line, it is advanced to the beginning of the next line. In overwrite mode, the cursor is advanced without altering the text in memory. In insert mode, spaces are inserted as the cursor is advanced.

Each of the following advances the cursor to the next occurrence of a specific character in the text memory. If the character is not found, the cursor is positioned at the beginning of the text memory. This makes it easy to cycle through the text memory to find all occurrences.

This key combination... advances the cursor to the next:

[SHIFT]+[CTRL]+[SPACE]	space
[SHIFT]+[CTRL]+[RETURN]	return character
[SHIFT]+[CTRL]+[.]	period
[SHIFT]+[CTRL]+[?]	question mark
[SHIFT]+[CTRL]+[!]	exclamation mark
[SHIFT]+[CTRL]+[D]	define command (INV-CTRL-D)
[SHIFT]+[CTRL]+[E]	end page command (INV-CTRL-E)
[SHIFT]+[CTRL]+[M]	block marker (INV-CTRL-M)
[SHIFT]+[CTRL]+[O]	omit command (INV-CTRL-O)
[SHIFT]+[CTRL]+[P]	page format command (INV-CTRL-P)
[SHIFT]+[CTRL]+[Q]	quit printing command (INV-CTRL-Q)
[SHIFT]+[CTRL]+[CAPS]	control character
[SHIFT]+[CTRL]+[INVERSE]	inverse video character

Other Methods of Moving the Cursor

[SHIFT]+[CTRL]+[S]	This advances the cursor to the next match with the search specification found in the text memory (The SELECT menu is used to define the search specification.) Question marks in the search specification are treated as wild card characters. This means that any character in the text will be considered a match to the question marks that are in the search specification. If the search specification is null, the cursor will be advanced one character in the text memory. If no match is found, the cursor is positioned at the beginning of the text memory. This allows a document to be easily cycled through using the search operation. The operation can also be carried out with [START] [S] [S].
--------------------	--

Printing the document also advances the cursor through the text memory. Documents can be printed to the screen (S:) or the screen editor (E:) and interrupted at any time with [BREAK]. When [BREAK] is pressed the text window is restored and the cursor will be positioned at the part of the document that was about to be printed. Doing this with a document that contains a page wait command is an easy way to find page breaks.

Cut and Paste Operations

Each of the following key commands causes part of the text in memory to be deleted and appended to a 2000 character delete buffer. The delete buffer is first cleared if the previous keystroke was not also one of these deletions. The contents of the delete buffer can be inserted into the text at the cursor position with

[SHIFT]+[INSERT] or [START] [I] [B].

These keys can be used to quickly "cut and paste" small portions of text within a document or from one document to another.

[SHIFT]+[DELETE] This deletes the line of text that the cursor is on. It can also be done with [START] [D] [L].

[CTRL]+[(] This deletes text from the beginning of the line, that the cursor is on to the cursor. It can also be done with [START] [D] [(].

[CTRL]+[)] This deletes text from the cursor to the end of the line that the cursor is on. It can also be done with [START] [D] [)].

[CTRL]+[SPACE] This deletes text from the cursor until the next space is found in the text memory. It can be used to delete words by first positioning the cursor on the space in front of the word that is to be deleted. The operation can also be done with [START] [D] [S].

[CTRL]+[RETURN] This deletes text from the cursor to the next return character in the text memory. It can also be done with [START] [D] [R].

Erase Operations

The following key commands cause text to be erased from the text memory. Text erased in this manner cannot be restored except by retyping it.

[BACK SPACE]	This moves the cursor one character to the left. In overwrite mode, this character is replaced with a space. In insert mode, the character is removed from the text memory. This means that in both modes the back space key undoes what the space bar does.
[CTRL]+[DELETE]	This removes the character at the cursor from the text memory.
[SHIFT]+[CTRL]+[DELETE]	This removes a contiguous block of spaces from the text memory beginning at the cursor character. If the cursor character is not a space, no action is taken. It can also be done with [START] [E] [S].
[SHIFT]+[CTRL]+[R]	This carries out the replacement operation and can be used to erase occurrences of the search specification when the replacement specification is null. It can also be done with [START] [S] [R].

Inserting Spaces at the Cursor

Each of the following key commands causes spaces to be inserted into the text memory at the cursor position without losing any of the text in memory.

[CTRL]+[INSERT]	This inserts one space at the cursor position. The cursor is positioned on the space that was inserted.
[SPACE] (in insert mode)	This inserts one space at the cursor position. The cursor is then advanced one character so that it remains on the same character that it was on originally.

- [TAB] (in insert mode) This inserts spaces at the cursor position as the cursor is advanced to the next tab stop. The cursor ends up on the same character that it was on originally.
- [SHIFT]+[CTRL]+[INSERT] This inserts approximately one screenful of spaces at the cursor position. If less than one screenful of space is available in the text memory, as many spaces as will fit are inserted. This provides the quickest method of entering text near the beginning of a large document by creating spaces which can be written over in overwrite mode. After the desired text is added, the extra spaces can be quickly removed with [SHIFT]+[CTRL]+[DELETE]. This operation can also be done with [START] [I] [S].
- [SHIFT]+[CTRL]+[R] This carries out the replacement operation, which can be used to insert spaces in the replacement specification at the cursor position when the search specification is null. It can also be done with [START] [S] [R].

Search and Replacement

- [SHIFT]+[CTRL]+[S] This advances the cursor to the next match with the search specification found in the text memory. (The SELECT menu is used to define the search specification.) Question marks in the search specification are treated as wild card characters, which means that any character in the text will be considered a match to the question marks that are in the search specification. If the search specification is null, the cursor will be advanced one character in the text memory. If no match is found, the cursor is positioned at the beginning of the text memory. This allows a document to be cycled through easily using the search operation. The search operation can also be carried out with [START] [S] [S].

[SHIFT]+[CTRL]+[R] This carries out the replacement operation. If the cursor is positioned at a portion of text that matches the search specification, that text is replaced with the replacement specification text. (The SELECT menu is used to define the search and replacement specifications.) If the cursor is not at a match position, a search operation is performed. The replacement operation supports the use of question marks as wild card characters. (See **Search and/or Replace** in **THE START MENU** for more details, page 26.) The replacement operation can also be done with [START] [S] [R].

The Macro Key Command

The macro key command permits the execution of up to 80 individual key commands with a single keystroke. Any keystroke may be included in the macro key definition except [BREAK], [START], [SELECT], [OPTION], [RESET], and those listed below which are used to start, end, and execute the definition.

[CTRL] + [ESC] This clears the previous macro key command definition and begins the new definition. Each subsequent keystroke is included in the new macro key command definition.

[SHIFT] + [ESC] If the process of defining the macro key command is in progress, this key command ends the definition. Otherwise, the current key command is executed.

Special Input Line Key Commands

The following key commands are available for use on the input line.

[CTRL]+[LEFT-ARROW] This moves the input cursor one character to the left. If the input cursor is at the beginning of the input line it will not be moved.

[CTRL]+[RIGHT-ARROW] This moves the input cursor one character to the right. If the input cursor is at the end of the current input it will not be moved.

[CTRL]+[INSERT] This causes a space to be inserted at the input cursor position.

[CTRL]+[DELETE]	This causes the character at the cursor to be removed from the input line. If there are no characters on the input line no action occurs.
[BACK SPACE]	This causes the character to the left of the cursor to be replaced with a space and the input cursor to be moved to this position. If the input cursor is positioned at the end of the current input the length of the input is decreased by one character. If there are no characters on the input line no action occurs.
[SHIFT]+[DELETE]	This causes the entire input line to be cleared. The input cursor is positioned at the beginning of the input line.
[SHIFT]+[INSERT]	This causes the contents of the input line to be inserted into the main text at the cursor position. The input is then terminated.
[CTRL]+[UP-ARROW]	This causes the cursor character in the main text to be copied up onto the input line at the input cursor position. Both the main text cursor and the input cursor are moved one character to the right, if possible.
[ESC]	This causes the next regular keystroke or key combination to produce a character on the input line. It is used to produce characters with keystrokes that normally carry out an operation. This key makes it possible to enter any of the 256 ASCII characters on the input line (including the return character).
[RETURN]	This causes the contents of the input line to be used as the response to the prompt.
[BREAK]	This causes the input to be terminated with no operation carried out. The default input (if any) remains as it was before the input began.

IMBEDDED PRINT CONTROL COMMANDS

Imbedded print control commands provide for extensive control of the printer while a document is being printed. These commands contain special command characters which are really inverse video control characters, most of which appear on the screen as underlined upper case letters. The exceptions are the four inverse video arrow characters, which have their normal appearance. As mentioned above, these command characters can be easily inserted into the text by first pressing the BREAK key.

The syntax of each command is shown below. Information enclosed within bent brackets (<>) describes the type of text that should appear. The bent brackets themselves should not be typed. The down left arrow (↵) is the symbol for the return character. When shown, it must be typed. Except for centering and right margin alignment, commands that are required to end with a return character do not cause the paper to be advanced when the document is printed.

Horizontal Alignment Commands

L <auto indented text> Use left margin
When used at the beginning of a line that would normally begin at the indent margin, the line is forced to begin at the left margin. Such lines are accurately displayed on the screen.

C <Text>↵ Center
This causes the text between this command character and the next return character to be centered between the left-most margin and the right margin.

R <Text>↵ Right margin alignment
This causes the text between this command character and the next return character to be positioned against the right margin.

<Text-1> C <Text-2>↵ These combinations of the above
<Text-1> R <Text-2>↵ horizontal alignment commands are
L <Text-1> C <Text-2>↵ supported. It is not possible to use
L <Text-1> R <Text-2>↵ both the centering and the right margin
alignment commands on the same line.

Page Control Commands

E ↵ End or eject page
This causes the current page to be ended. Printing then continues with the next page.

P <Lf>,<In>,<Rg>,<Hd>,...Page format parameters
 This permits any of the ten page format parameters to be changed. (See **Page Format Parameters** in **THE SELECT MENU**, pages 19-20.)

Defining Headers and Footers

H <Header text>␣ Define header text
 This defines the text that will be printed on the header line on subsequent pages. If a header is to appear on the first page, the header must be defined in the document before the first line of text. If the header definition does not contain any horizontal alignment commands, it will be printed at the left-most margin.

F <Footer text>␣ Define footer text
 This defines the text that will be printed on the footer line of the current and subsequent pages. If the footer definition does not contain any horizontal alignment commands, it will be printed at the left-most margin.

Page and Item Number Commands

S N <#>␣ Set current page number
 The page number is set to 1 when the printing operation begins and is incremented with each new page. This command can be used to change the current page number. It may be assigned any value from 0 to 9999. The current page number is printed any time the imbedded command character **N** occurs within the text.

S X <#>␣ Set item number
 The item number is initially set to one when the printing operation begins. This command is used to change the current item number. It may be assigned any value from 0 to 9999. The item number is printed and incremented each time the imbedded command character **X** occurs within the text.

<Text> N <Text> Print page number
 This causes the current page number to be printed. The command can be used within a header or footer definition to number the pages of a document automatically.

<Text> X <Text> Print item number
 This causes the item number to be printed and incremented. It is useful for automatic numbering of items in a document.

Macro Printer Codes

D ↑ <Printer codes>↵ Define arrows
D ↓ <Printer codes>↵ These define the printer codes that will be substituted for the various inverse video arrow characters, when they are imbedded within a line of text. Definitions may be up to 32 characters in length.
D ← <Printer codes>↵
D → <Printer codes>↵

D I <Printer codes>↵ Define top of page initialization
 This defines the top of page printer initialization codes. These codes are sent to the printer, as each new page begins. The result is that pages are printed properly even when a document is not printed from the beginning or when the printer is turned off between pages. The codes are also sent to the printer any time the imbedded command character I occurs in the text. Such a definition may be up to 32 characters in length.

D <INV-SPACE><Codes>↵ Define inverse video text
 This defines the printer codes that will be sent to the printer after a regular video text character is processed which is followed in the text memory by an inverse video text character. Each subsequent inverse video text character that is processed is converted to regular video before it is sent to the printer. When inverse video is given a null definition, the subsequent inverse video characters are sent to the printer without any extra codes or modifications.

D <REG-SPACE><Codes><V

Define exit from inverse video text
This defines the printer codes that will be sent to the printer after an inverse video text character is processed which is followed in the text memory by a regular video text character or which is at the end of a line. This process is disabled when inverse video (see above) is given a null definition.

<Text> ↑ <Text>

<Text> ↓ <Text>

<Text> ← <Text>

<Text> → <Text>

Arrow codes to printer

These cause the printer control codes, previously defined for these inverse video arrow characters to be sent to the printer verbatim.

<Text> I <Text>

Top of page codes to printer

This causes the previously defined printer top of page control codes to be sent to the printer before the next character is processed. These codes are automatically sent to the printer when each new page begins.

Print Mode Toggles

U <Text to underline> U

Underline toggle

Each time this command character is processed the underline mode is toggled. When on, a back space command (ASCII 8) and an underline character (ASCII 95) are sent to the printer after each of the subsequent characters. When off, no extra codes are sent to the printer. This causes the text that is delimited by two of these command characters to be underlined.

W <V

Wait at end of page toggle

This toggles the page wait option. Without it, the print operation prints one page after another without halting. With it, printing waits at the end of each page until a key on the keyboard is pressed. This permits single sheet printing.

Miscellaneous Print Control Commands

- I** <Text file spec>␣ Include or insert load a file
This causes the specified file to be included in the printing of the document. This command, and the text above it, are cleared from the text memory and the contents of the specified file are inserted in front of the text that was left in the text memory. Printing is then resumed from the beginning of the text memory. This may be used at the end of each file in a chain of files that are to be printed together as one document to specify the next file in the chain. This may also be used in a batch file to specify all the files that are to be printed together.
- I V** <Verbatim file sp>␣ Include a verbatim file
This causes the contents of the specified file to be transferred to the printer verbatim. This is useful for printing formatted text files, graphics files or files used to redefine the shapes of the printer's characters. The contents of the text memory is not altered by this command.
- O** <Remark or comment>␣ Omit when printing
The text from the omit command character to the next return character is not printed. This can be used to imbed remarks within the text that are not to be part of the printed document.
- <Text> **Q** <Text> Quit until a key is pressed
This causes the printing process to be interrupted, until a key on the keyboard is pressed to continue.
- V** <Printer codes>␣ Verbatim to printer
This causes the characters between this command character and the next return character to be sent to the printer verbatim.
- [CTRL]+[COMMA] Hard space character
This produces a character that has the same appearance on the screen as a false space. It is printed as a space, but forces adjacent items to be printed on the same line when word wrap is on.

SUGGESTIONS AND HINTS

Methods of Inserting Text Quickly

While insert mode does allow text to be inserted into a document as it is typed, the process can be very slow near the beginning of a large document. There are alternate methods of inserting such text.

The first method is to position the cursor where the text is to be inserted and then to create a screenful of spaces with either [SHIFT]+[CTRL]+[INSERT] or [START] [I] [S]. The desired text can then be entered very quickly in overwrite mode. Any extra spaces that are left can be removed quickly using [SHIFT]+[CTRL]+[DELETE] or [START] [E] [S].

The second method is to first enter the text at the end of the document that is in memory and then move it to the desired location. If less than 2000 characters need to be moved, they can first be deleted to the delete buffer and then inserted at the desired location. (See **Delete Text to Buffer**, page 23, and **Insert Operations**, page 25, in **THE START MENU** or **Cut and Paste Operations** in **KEY COMMANDS**, page 32.) If more than 2000 characters need to be moved, they can be marked as a block and then moved to any other part of the document in the text memory. (See **Block Operations** in **THE START MENU**, page 23.)

The third method is to enter the text that is to be inserted in a separate file. Once saved, the contents of the file can be inserted directly into any other file. (See **Load File** in **THE OPTION MENU**, page 15, or **Insert Operations** in **THE START MENU**, page 25.) This file can also be designated to be inserted into any other file at the time the document is printed. (See how to include a file in **Miscellaneous Print Control Commands** in **IMBEDDED PRINT CONTROL COMMANDS**, page 41.)

Establishing a Pseudo Printer Driver

Rewrite does not make use of printer drivers as such. Instead, printer control codes are defined within each document. Frequently-used printer code definitions for use with different printers may be saved in separate files. These files may then be inserted directly into the appropriate text files. (See **Load File** in **THE OPTION MENU**, page 15, or **Insert Operations** in **THE START MENU**, page 25.) They may also be designated to be inserted at the time the document is printed. Printer initialization codes (such as for defining a new character set for dot matrix printers) can also be included from a verbatim file. (See **Miscellaneous Print Control Commands** in **IMBEDDED PRINT CONTROL COMMANDS**, page 41, for instructions on how to include text and verbatim files.)

THE REWRITE WORDPROCESSOR Version 2.4

By: Robert Ertl - Copyright 1987

5121 - 16th Street Racine, WI

Cursor Movement:

Up one line.....[CTRL] + [UP]
Down one line.....[CTRL] + [DOWN]
Left one character.....[CTRL] + [LEFT]
Right one character.....[CTRL] + [RIGHT]
Up ten lines.....[SHIFT] + [CTRL] + [UP]
Down ten lines.....[SHIFT] + [CTRL] + [DOWN]
To left edge.....[SHIFT] + [CTRL] + [()][br/>To right edge.....[SHIFT] + [CTRL] + [)]
To top of text.....[CTRL] + [CLEAR]
To bottom of text.....[SHIFT] + [CTRL] + [CLEAR]
To top of screen.....[SHIFT] + [CTRL] + [CLEAR]

Device Operations:

Display disk directory.....[OPTION] [D]
Erase disk file.....[OPTION] [E]
Format disk.....[OPTION] [F]
Insert-Load at cursor.....[OPTION] [L] [I] or
[START] [I] [L]
Append-Load file.....[OPTION] [L] [A]
Overwrite-Load from top.....[OPTION] [L] [T]
Overwrite-Load at cursor.....[OPTION] [L] [C]
Print document.....[OPTION] [P]
Rename file.....[OPTION] [R]
Save all text.....[OPTION] [S] [A]
Save marked block.....[OPTION] [S] [B] or
[START] [B] [S]
Save from cursor down.....[OPTION] [S] [D]
Save from cursor up.....[OPTION] [S] [U]
Save word processor.....[OPTION] [S] [W]
Unprotect file.....[OPTION] [U]
View/Verify file.....[OPTION] [V]
• Write-Protect file.....[OPTION] [W]

Check Free Memory:

Check free memory.....[SELECT] [F]

Special Keys During Input:

Insert a space.....[CTRL] + [INSERT]
Erase character at cursor.....[CTRL] + [DELETE]
Clear input line.....[SHIFT] + [DELETE]
Erase character left of cursor..[BACK S]
Move cursor left.....[CTRL] + [LEFT]
Move cursor right.....[CTRL] + [RIGHT]
Copy from main text.....[CTRL] + [UP]
Insert into main text.....[SHIFT] + [INSERT]
Enter input.....[RETURN]
Special character with next key.[ESC]
Abort input (default unchanged) [BREAK]
Note: Input defaults are suppressed when a function
key is depressed as the input prompt appears.

Macro Key: (Up to 80 keystrokes repeated)

Begin macro key definition.....[CTRL] + [ESC]
End macro key definition.....[SHIFT] + [ESC]
Execute defined macro key.....[SHIFT] + [ESC]

Advance Cursor to next...

Tab stop.....[TAB]
Space.....[SHIFT] + [CTRL] + [SPACE]
Return character.....[SHIFT] + [CTRL] + [RETURN]
Period.....[SHIFT] + [CTRL] + [.]
Question mark.....[SHIFT] + [CTRL] + [?]
Exclamation point.....[SHIFT] + [CTRL] + [!]
Define command.....[SHIFT] + [CTRL] + [D]
End of page command.....[SHIFT] + [CTRL] + [E]
Block marker.....[SHIFT] + [CTRL] + [M]
Omit command.....[SHIFT] + [CTRL] + [O]
Page format command.....[SHIFT] + [CTRL] + [P]
Quit printing command.....[SHIFT] + [CTRL] + [Q]
Control character.....[SHIFT] + [CTRL] + [CAPS]
Inverse video character.....[SHIFT] + [CTRL] + [INVERSE]

Choose Screen Settings:

Increment screen color number...[SELECT] [D] [>]
Decrement screen color number...[SELECT] [D] [<]
Reverse screen luminance.....[SELECT] [D] [L]
Narrow screen display.....[SELECT] [D] [N]
Widen screen display.....[SELECT] [D] [W]
Set page margins & format.....[SELECT] [F]
Activate proofreading screen...[SELECT] + [P] [RETURN]
Toggle Auto-Page Format.....[SELECT] [T] [A]
Toggle false spaces.....[SELECT] [T] [F] or
[SHIFT] + [CTRL] + [F]
Toggle guide bar.....[SELECT] [T] [G] or
[SHIFT] + [CTRL] + [G]
Toggle word-wrap.....[SELECT] [T] [W] or
[SHIFT] + [CTRL] + [W]
Choose error bell.....[SELECT] [T] [E] or
[CTRL] + [2]

Choose Other Settings & Modes:

Set keystrokes before save.....[SELECT] [K]
Set linefeed ASCII (0=none).....[SELECT] [L]
Toggle insert mode.....[SELECT] [T] [I] or
[SHIFT] + [CTRL] + [I]
Toggle regular/inverse video...[INVERSE-KEY]
Toggle upper/lower case entry...[CAPS]
Uppercase entry lock.....[SHIFT] + [CAPS]
Control character entry lock....[CTRL] + [CAPS]
Set tab stop at cursor.....[SHIFT] + [TAB]
Clear tab stop at cursor.....[CTRL] + [TAB]
Clear all tab stops.....[SHIFT] + [CTRL] + [TAB]
Special character with next key.[ESC]
Command character with next key.[BREAK]

Block Operations:

Mark block beginning or end....[SELECT] [B] [M] or
[BREAK] [M] or
[BREAK] + [M]
Delete block to buffer.....[START] [B] [D] or
[START] [D] [B]
Erase block.....[START] [B] [E] or
[START] [E] [B]
Move block to cursor.....[START] [B] [M]
Save block.....[START] [E] [S] or
[OPTION] [S] [B]
Unmark block.....[START] [E] [U] or
[SELECT] [B] [U]

Delete Operations:

Delete block *.....[START] [D] [B] or
[START] [B] [D]
Delete character.....[START] [D] [C]
Delete line.....[START] [D] [L] or
[SHIFT] + [DELETE]
Delete left of cursor.....[START] [D] [()] or
[CTRL] + [()
Delete right of cursor.....[START] [D] [)] or
[CTRL] + [)]
Delete until space.....[START] [D] [S] or
[CTRL] + [SPACE]
Delete until return.....[START] [D] [R] or
[CTRL] + [RETURN]
Zap the delete buffer.....[START] [D] [Z]
Notes: All deletions are appended to the delete buffer
* Delete block clears the buffer first

Recalling Deleted Text:

Insert buffer at cursor.....[START] [I] [B] or
[SHIFT] + [INSERT]

Erase Operations:

Erase character left of cursor.....[BACK S]
Erase cursor character.....[CTRL] + [DELETE]
Erase all text.....[START] [E] [A]
Erase marked block.....[START] [E] [B] or
[START] [B] [E]
Erase text from cursor up.....[START] [E] [U]
Erase text from cursor down.....[START] [E] [D]
Erase spaces after cursor.....[START] [E] [S] or
[SHIFT] + [CTRL] + [DELETE]
Note: Erased text cannot be recalled!

Inserting Spaces at the Cursor:

Insert one space.....[CTRL] + [INSERT]
Insert one space.....[SPACE] (in insert mode)
Insert spaces to tab stop.....[TAB] (in insert mode)
Insert screenful of space.....[START] [I] [S] or
[SHIFT] + [CTRL] + [INSERT]
Insert all available space.....[START] [I] [A]

Search and Replacement:

Define search specification.....[SELECT] [S]
("?" is a wild card character)
Define replace specification.....[SELECT] [R]
("?" is a wild card character)
Search for search specification.....[START] [S] [S] or
[SHIFT] + [CTRL] + [S]
Attempt one replacement.....[START] [S] [R] or
[SHIFT] + [CTRL] + [R]
Do global search & replace.....[START] [S] [G]
beginning at cursor

Text Conversion

Convert to uppercase.....[START] [T] [U]
Convert to lowercase.....[START] [T] [L]
Convert to regular video.....[START] [T] [R]
Convert to inverse video.....[START] [T] [I]

Imbedded Print Control Commands:

Horizontal Text Alignment
Begin at left margin.....L <Text at left margin>
Center text.....C <Text to be centered>
Right margin alignment.....R <Text to end at right>
Text-1 reg, Center Text-2.....<Text-1> C <Text-2>
Text-1 reg, Text-2 to RM.....<Text-1> R <Text-2>
Text-1 at LM, Center Text-2.....<Text-1> C <Text-2>
Text-1 at LM, Text-2 to RM.....<Text-1> R <Text-2>

Page control

End or eject page.....E V
Set page format parameters.....P <Lf>, <In>, <Rg>, <Ls>,
<Lf> = Left margin position....(0...195)
<In> = Indent margin position.....(0...195)
<Rg> = Right margin position.....(2...213)
<Ls> = Line spacing.....(0...255)
<Ps> = Paragraph spacing.....(0...255)
<Hd> = Header line.....(0...255)
<Tp> = Top line.....(0...255)
<Bt> = Bottom line.....(0...255)
<Ft> = Footer line.....(0...255)
<Ln> = Length of page.....(0...255)
Example: P 10,,80 sets the left & right margins

Defining Headers & Footers

Define header.....H <Header text>
Define footer.....F <Footer text>
These may include other print control commands.

Page & Item Numbers

Set page no. (0...9999).....S N <#>
Set item no. (0...9999).....S X <#>
Print current page no.....<Text> N <Text>
Print & incr. item no.....<Text> X <Text>

Macro Printer Codes

Define up-arrow codes.....D ↑ <Printer codes>
Define down-arrow codes.....D ↓ <Printer codes>
Define left-arrow codes.....D ← <Printer codes>
Define right arrow codes.....D → <Printer codes>
Define Top of page codes.....D I <Printer init codes>
Define inverse video codes.....D <INV-SPACE> <Pr. Codes>
Define regular video codes.....D <REG-SPACE> <Pr. Codes>
Print up-arrow codes.....<Text> ↑ <Text>
Print down-arrow codes.....<Text> ↓ <Text>
Print left-arrow codes.....<Text> ← <Text>
Print right-arrow codes.....<Text> → <Text>
Print top of page codes.....<Text> I <Text>
Print inverse video codes.....<Regular Text> <Inverse text>
Print regular video codes.....<Inverse Text> <Regular text>

Miscellaneous

Include a text file.....I <Text file specification>
Include verbatim file.....I V <Verbatim file spec>
Omit comments & remarks.....O <Text to be omitted>
Quit printing until key.....<Text> Q <Text>
Underline the text.....<Text> U <Text to underline>
Send pr. codes verbatim.....V <Printer codes>
Toggle end-of-page-wait.....W
Hard Space character.....[CTRL] + [,]

Notes: Print control commands are inverse-control-characters

To enter codes quickly, push [BREAK] first.

Unless otherwise noted, print control commands should be the first character(s) on the line.

"\n" is the return character.